

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-184699

(43)Date of publication of application : 09.07.1999

(51)Int.Cl.

G06F 9/44

G06F 13/00

G06F 15/16

(21)Application number : 09-324945

(71)Applicant : INTERNATL BUSINESS MACH CORP <IBM>

(22)Date of filing : 26.11.1997

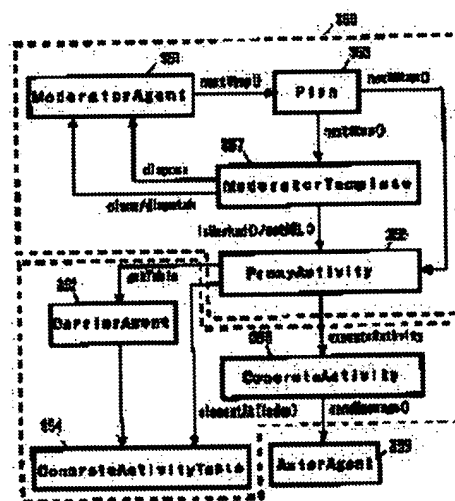
(72)Inventor : SUZUKI TOSHIHIRO
MINAMI KAZUHIRO

(54) METHOD FOR EXECUTING MOBILE OBJECT GROUP AND STORAGE MEDIUM STORING MOBILE OBJECT GROUP

(57)Abstract:

PROBLEM TO BE SOLVED: To suppress the load of a network to the minimum at the time of moving a mobile agent for executing a job while moving to places distributed over a network.

SOLUTION: A concrete activity 358 for defining a job to be executed at each place is moved to each place putting on a carrier agent 352. On the other hand, a main body 350 of a mobile object group holds a proxy activity 355 for sending the instruction of execution to the concrete activity 358 held at the carrier agent. This main body 350 of mobile the object group is circulated through respective places and the instruction of execution is sent to the concrete activity 358 moving to each place.



LEGAL STATUS

[Date of request for examination] 28.12.1999

[Date of sending the examiner's decision of rejection] 03.07.2002

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection] 2002-18870

[Date of requesting appeal against examiner's decision of rejection] 27.09.2002

[Date of extinction of right]

BEST AVAILABLE COPY

Copyright (C); 1998,2003 Japan Patent Office

(51)Int.Cl.⁶

識別記号

FI

G 0 6 F 9/44
13/00
15/165 5 2
3 5 1
3 7 0C 0 6 F 9/44
13/00
15/165 5 2
3 5 1 F
3 7 0 N

審査請求 未請求 請求項の数 8 OL (全 37 頁)

(21)出願番号 特願平9-324945

(22)出願日 平成9年(1997)11月26日

(71)出願人 390009531

インターナショナル・ビジネス・マシーン
ズ・コーポレーションINTERNATIONAL BUSIN
ESS MACHINES CORPO
RATIONアメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)

(72)発明者 鈴木 俊 宏

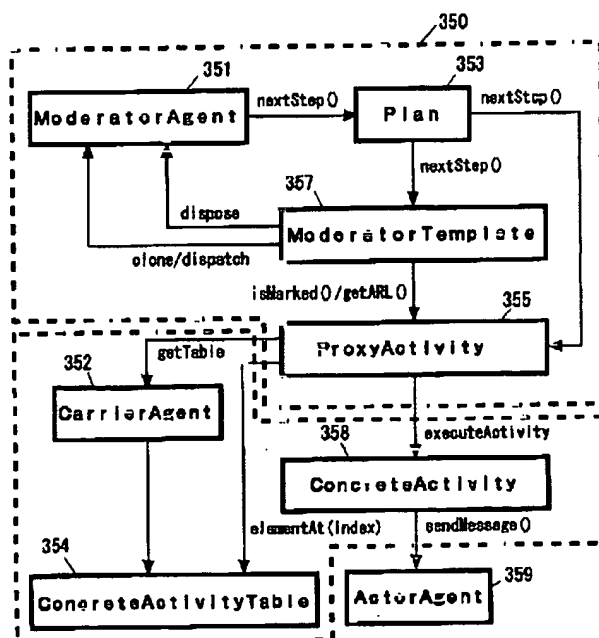
神奈川県大和市下鶴間1623番地14 日本ア
イ・ピー・エム株式会社 大和事業所内

(74)代理人 弁理士 坂口 博 (外1名)

最終頁に続く

(54)【発明の名称】 移動オブジェクト群の実行方法、及び移動オブジェクト群を格納した記憶媒体

(57)【要約】

【課題】 ネットワークに分散するブレースに移動し、
ジョブを実行する移動エージェントの移動時におけるネ
ットワークの負荷を最小限に抑える。【解決手段】 各ブレースで実行するジョブを定義するコ
ンクリート・アクティビティ358をキャリア・エー
ジェント352に乗せて各ブレースに移動させる。この一
方、移動オブジェクト群本体350は、キャリア・エー
ジェントに保持されたコンクリート・アクティビティ3
58に実行の指示を送出するプロキシ・アクティビ
ティ355を保持する。この移動オブジェクト群本体35
0が各ブレースを巡回し、各ブレースに移動しているコ
ンクリート・アクティビティ358に実行の指示を送出
していく。

【特許請求の範囲】

【請求項1】ネットワーク上に分散する異なるプレースにおいて異なるジョブを実行する方法であって、

(a) 第1のプレースで実行するジョブを定義する第1のコンクリート・アクティビティを保持する第1のキャリア・エージェントを生成する段階と、

(b) 第2のプレースで実行するジョブを定義する第2のコンクリート・アクティビティを保持する第2のキャリア・エージェントを生成する段階と、

(c) 前記第1のキャリア・エージェントを特定する情報と前記第2のキャリア・エージェントを特定する情報とを含み、前記第1のプレースにおいて前記第1のキャリア・エージェントに対し、前記第1のコンクリート・アクティビティにジョブの実行を指示し、前記第2のプレースにおいて前記第2のキャリア・エージェントに対し、前記第2のコンクリート・アクティビティにジョブの実行を指示する移動オブジェクト群本体を生成する段階と、

(d) 前記第1のキャリア・エージェントを前記第1のプレースに移動させる段階と、

(e) 前記第2のキャリア・エージェントを前記第2のプレースに移動させる段階と、

(f) 前記移動オブジェクト群本体を前記第1のプレースに移動させる段階と、

(g) 前記第1のプレースにおいて、前記移動オブジェクト群本体が前記第1のキャリア・エージェントを特定し、前記第1のコンクリート・アクティビティにジョブの実行を指示することにより、前記第1のコンクリート・アクティビティのジョブを実行する段階と、

(h) 前記移動オブジェクト群本体を前記第2のプレースに移動させる段階と、

(i) 前記第2のプレースにおいて、前記移動オブジェクト群本体が前記第2のキャリア・エージェントを特定し、前記第2のコンクリート・アクティビティにジョブの実行を指示することにより、前記第2のコンクリート・アクティビティのジョブを実行する段階と、
を含む移動オブジェクト群の実行方法。

【請求項2】ネットワーク上に分散する少なくとも1つのプレースにおいてジョブを実行する方法であって、

(a) 前記プレースで実行するジョブを定義するコンクリート・アクティビティを保持するキャリア・エージェントを生成する段階と、

(b) 前記キャリア・エージェントを特定する情報を含み、前記プレースにおいて前記キャリア・エージェントに対し、前記コンクリート・アクティビティにジョブの実行を指示する移動オブジェクト群本体を生成する段階と、

(c) 前記キャリア・エージェントを前記プレースに移動させる段階と、

(d) 前記移動オブジェクト群本体を前記プレースに移

動させる段階と、

(e) 前記プレースにおいて、前記移動オブジェクト群本体が前記キャリア・エージェントを特定し、前記コンクリート・アクティビティにジョブの実行を指示することにより、前記コンクリート・アクティビティのジョブを実行する段階と、
を含む移動オブジェクト群の実行方法。

【請求項3】ネットワーク上に分散する少なくとも1つのプレースにおいてジョブを実行する方法であって、

(a) 前記プレースで実行するジョブを定義するコンクリート・アクティビティを保持するキャリア・エージェントを前記プレースに移動させる段階と、

(b) 前記キャリア・エージェントを特定する情報を含み、前記プレースにおいて前記キャリア・エージェントに対し、前記コンクリート・アクティビティにジョブの実行を指示する移動オブジェクト群本体を前記プレースに移動させる段階と、

(c) 前記プレースにおいて、前記移動オブジェクト群本体が前記キャリア・エージェントを特定し、前記コンクリート・アクティビティにジョブの実行を指示することにより、前記コンクリート・アクティビティのジョブを実行する段階と、
を含む移動オブジェクト群の実行方法。

【請求項4】ネットワーク上に分散する異なるプレースにおいて異なるジョブを実行する移動オブジェクト群の少なくとも一部を格納する記憶媒体であって、

(a) 第1のプレースで実行するジョブを定義する第1のコンクリート・アクティビティを保持する第1のキャリア・エージェントと、

(b) 第2のプレースで実行するジョブを定義する第2のコンクリート・アクティビティを保持する第2のキャリア・エージェントと、

(c) 前記第1のキャリア・エージェントを特定する情報と前記第2のキャリア・エージェントを特定する情報とを含み、前記第1のプレースにおいて前記第1のキャリア・エージェントに対し、前記第1のコンクリート・アクティビティにジョブの実行を指示し、前記第2のプレースにおいて前記第2のキャリア・エージェントに対し、前記第2のコンクリート・アクティビティにジョブの実行を指示する移動オブジェクト群本体と、
を格納する記憶媒体。

【請求項5】ネットワーク上に分散するプレースにおいてジョブを実行する移動オブジェクト群の少なくとも一部を格納する記憶媒体であって、

(a) プレースで実行するジョブを定義するコンクリート・アクティビティを保持するキャリア・エージェントと、

(b) 前記キャリア・エージェントを特定する情報を含み、前記プレースにおいて前記キャリア・エージェントに対し、前記コンクリート・アクティビティにジョブの

実行を指示する移動オブジェクト群本体と、
を格納する記憶媒体。

【請求項6】ネットワーク上に分散する異なるプレースにおいてメッセージを送出する移動オブジェクト群の少なくとも一部を格納する記憶媒体であって、

(a) 第1のプレースで実行するジョブを定義する第1のコンクリート・アクティビティを保持する第1のキャリア・エージェントを特定する情報を含み、前記第1のプレースにおいて前記第1のキャリア・エージェントに対し、前記第1のコンクリート・アクティビティにジョブの実行を指示する第1のプロキシ・アクティビティと、

(b) 第2のプレースで実行するジョブを定義する第2のコンクリート・アクティビティを保持する第2のキャリア・エージェントを特定する情報を含み、前記第2のプレースにおいて前記第2のキャリア・エージェントに対し、前記第2のコンクリート・アクティビティにジョブの実行を指示する第2のプロキシ・アクティビティと、

(c) 前記第1のプロキシ・アクティビティと前記第2のプロキシ・アクティビティの実行順序を定義する順序リストと、

を格納する記憶媒体。

【請求項7】ネットワーク上に分散するプレースにおいて、前記プレースに到着したプロキシ・アクティビティを含む移動オブジェクト群本体の指示にตอบสนองしてジョブを実行する移動オブジェクトを格納する記憶媒体であって、

(a) 前記プレースで実行するジョブを定義するコンクリート・アクティビティと、

(b) 前記プロキシ・アクティビティが前記コンクリート・アクティビティを特定するために前記プロキシ・アクティビティに提供されるコンクリート・アクティビティ・テーブルと、

(c) 前記プロキシ・アクティビティからジョブの実行命令メッセージを受領するためのメソッドと、
を格納する記憶媒体。

【請求項8】ネットワーク上に分散するプレースにおいて、前記プレースに到着したプロキシ・アクティビティを含む移動オブジェクト群本体の指示にตอบสนองしてジョブを実行する移動オブジェクトを格納する記憶媒体であって、

(a) 前記プレースで実行するジョブを定義するコンクリート・アクティビティと、

(b) 前記プロキシ・アクティビティからジョブの実行命令メッセージを受領するためのメソッドと、
を格納する記憶媒体。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、分散コンピュータ環

境にあって、ネットワーク上に存在する遠隔サーバに移動し、該遠隔サーバにおいてインストラクションを形成する移動エージェント技術に関し、より詳しくは、移動エージェントを遠隔サーバに移動させる方式に関する。

【0002】

【従来の技術】従来より、分散コンピュータ環境において、ネットワーク上に存在するサーバに移動し、インストラクションを形成する移動エージェント（モバイルエージェント）技術が存在する（特開平7-182174号公報(USP5,603,031)、特表平7-509799号公報(国際出願番号:PCT/US94/07397、国際公開番号:W097/02219)、「日経コミュニケーション別冊、最新インターネットテクノロジー」日経BP社pp104-117、西田 文比古、藤原 進 著等）。

【0003】かかる移動エージェントは、「移動エージェント」と「プレース」という2つの基本要素を基に構成される。この移動エージェントは、自分自身の内部状態を保持したままネットワーク上に存在するプレース間を移動することができる。そして、このプレースにおいて他のエージェント（移動エージェントまたは非移動エージェント）とコンタクトし、必要なサービスを受けることができる。プレースは、ネットワーク上に存在するサーバによって提供されているエージェントの移動場所であり、エージェント間のコンタクトをサポートし、また、ハードウェア及びプラットフォームの違いを吸収している。

【0004】このような移動エージェント技術は、社内会議のセッティングを社員のスケジュールや会議室の予約状況に応じてダイナミックに調整したり、ネットワーク上に分散する所望の情報を入手する等、これまで人手によってなされていた作業をこの移動エージェントが代行することを可能にしている。

【0005】また、本願の出願時においては公知ではないが、本願出願人が平成9年4月10日に出願した特開平9-92091号公報のように、移動エージェントの複雑な行動を基本的行動パターンに分類し、その組合せによって、移動エージェントの所望の行動パターンを定義できるテンプレート（モデレータ・テンプレート）を提供し、移動エージェントが渡り歩く各プレースにおいてリクエストを出すアクティビティを、プレデセッサ（前任者：前のプレースで仕事をするアクティビティ）とサクセッサ（承継者：次のプレースで仕事をするアクティビティ）として管理する技術が存在する。

【0006】しかし、この技術を使用すると、各プレースにおいて実行するジョブを規定するアクティビティの全てを保持した状態で種々のプレースを渡り歩くこととなり、移動の際の情報量が多く、ネットワークへの負担が過大になり、各プレースにおいても必要以上の記憶域を占有してしまうという問題点があった（アクティビティは、他のオブジェクトに比べ非常に重い）。

【0007】また、予め決定された協調アルゴリズム（プラン）にのみ、そのモデルが適用され、移動エージェントが放たれた後、移動途中で実行するジョブを組み替えたり、新たに追加するような柔軟性のある行動は取れなかった。

【0008】

【発明が解決しようとする課題】本発明の一つの目的は、ネットワークに分散するプレースに移動し、ジョブを実行する移動エージェントの移動時におけるネットワークの負荷を最小限に抑えることにある。

【0009】本発明の一つの目的は、各プレースの資源の占有を最小限に抑えることのできる移動エージェントを提供することにある。

【0010】本発明の他の一つの目的は、移動途中で柔軟に実行するジョブを変更することのできる移動エージェントを提供することにある。

【0011】本発明の他の一つの目的は、移動エージェントを放ってから結果が返ってくるまでに要する処理時間を短縮することにある。

【0012】

【課題を解決するための手段】本発明においては、各プレースで実行するジョブを定義するコンクリート・アクティビティをキャリア・エージェントに乗せて各プレースに移動させる。この一方、移動オブジェクト群本体は、キャリア・エージェントに保持されたコンクリート・アクティビティに実行の指示を送出するプロキシ・アクティビティを保持する。この移動オブジェクト群本体が各プレースを巡回し、各プレースに移動しているコンクリート・アクティビティに実行の指示を送出していく。これにより、巡回する全てのプレースで実行するコンクリート・アクティビティを保持して各プレースを渡り歩くよりも、送信すべき情報量が少なく済み、ネットワークの負荷を抑えることができ、高速な移動により最終結果を受領する時間が短縮できる。また、各プレースで占有する記憶域も少なくなる。

【0013】本発明の一態様においては、ネットワーク上に分散する異なるプレースにおいて異なるジョブを実行する方法であって、(a) 第1のプレースで実行するジョブを定義する第1のコンクリート・アクティビティを保持する第1のキャリア・エージェントを生成する段階と、(b) 第2のプレースで実行するジョブを定義する第2のコンクリート・アクティビティを保持する第2のキャリア・エージェントを生成する段階と、(c) 前記第1のキャリア・エージェントを特定する情報と前記第2のキャリア・エージェントを特定する情報とを含み、前記第1のプレースにおいて前記第1のキャリア・エージェントに対し、前記第1のコンクリート・アクティビティにジョブの実行を指示し、前記第2のプレースにおいて前記第2のキャリア・エージェントに対し、前記第2のコンクリート・アクティビティにジョブの実行

を指示する移動オブジェクト群本体を生成する段階と、(d) 前記第1のキャリア・エージェントを前記第1のプレースに移動させる段階と、(e) 前記第2のキャリア・エージェントを前記第2のプレースに移動させる段階と、(f) 前記移動オブジェクト群本体を前記第1のプレースに移動させる段階と、(g) 前記第1のプレースにおいて、前記移動オブジェクト群本体が前記第1のキャリア・エージェントを特定し、前記第1のコンクリート・アクティビティにジョブの実行を指示することにより、前記第1のコンクリート・アクティビティのジョブを実行する段階と、(h) 前記移動オブジェクト群本体を前記第2のプレースに移動させる段階と、(i) 前記第2のプレースにおいて、前記移動オブジェクト群本体が前記第2のキャリア・エージェントを特定し、前記第2のコンクリート・アクティビティにジョブの実行を指示することにより、前記第2のコンクリート・アクティビティのジョブを実行する段階と、を含む移動オブジェクト群の実行方法が提供される。

【0014】本発明の他の一態様においては、ネットワーク上に分散する少なくとも1つのプレースにおいてジョブを実行する方法であって、(a) 前記プレースで実行するジョブを定義するコンクリート・アクティビティを保持するキャリア・エージェントを生成する段階と、(b) 前記キャリア・エージェントを特定する情報を含み、前記プレースにおいて前記キャリア・エージェントに対し、前記コンクリート・アクティビティにジョブの実行を指示する移動オブジェクト群本体を生成する段階と、(c) 前記キャリア・エージェントを前記プレースに移動させる段階と、(d) 前記移動オブジェクト群本体を前記プレースに移動させる段階と、(e) 前記プレースにおいて、前記移動オブジェクト群本体が前記キャリア・エージェントを特定し、前記コンクリート・アクティビティにジョブの実行を指示することにより、前記コンクリート・アクティビティのジョブを実行する段階と、を含む移動オブジェクト群の実行方法が提供される。

【0015】本発明の他の一態様においては、ネットワーク上に分散する少なくとも1つのプレースにおいてジョブを実行する方法であって、(a) 前記プレースで実行するジョブを定義するコンクリート・アクティビティを保持するキャリア・エージェントを前記プレースに移動させる段階と、(b) 前記キャリア・エージェントを特定する情報を含み、前記プレースにおいて前記キャリア・エージェントに対し、前記コンクリート・アクティビティにジョブの実行を指示する移動オブジェクト群本体を前記プレースに移動させる段階と、(c) 前記プレースにおいて、前記移動オブジェクト群本体が前記キャリア・エージェントを特定し、前記コンクリート・アクティビティにジョブの実行を指示することにより、前記コンクリート・アクティビティのジョブを実行する段階

と、を含む移動オブジェクト群の実行方法が提供される。

【0016】本発明の他の一態様においては、ネットワーク上に分散する異なるプレースにおいて異なるジョブを実行する移動オブジェクト群の少なくとも一部を格納する記憶媒体であって、(a)第1のプレースで実行するジョブを定義する第1のコンクリート・アクティビティを保持する第1のキャリア・エージェントと、(b)第2のプレースで実行するジョブを定義する第2のコンクリート・アクティビティを保持する第2のキャリア・エージェントと、(c)前記第1のキャリア・エージェントを特定する情報と前記第2のキャリア・エージェントを特定する情報とを含み、前記第1のプレースにおいて前記第1のキャリア・エージェントに対し、前記第1のコンクリート・アクティビティにジョブの実行を指示し、前記第2のプレースにおいて前記第2のキャリア・エージェントに対し、前記第2のコンクリート・アクティビティにジョブの実行を指示する移動オブジェクト群本体と、を格納する記憶媒体が提供される。

【0017】本発明の他の一態様においては、ネットワーク上に分散するプレースにおいてジョブを実行する移動オブジェクト群の少なくとも一部を格納する記憶媒体であって、(a)プレースで実行するジョブを定義するコンクリート・アクティビティを保持するキャリア・エージェントと、(b)前記キャリア・エージェントを特定する情報を含み、前記プレースにおいて前記キャリア・エージェントに対し、前記コンクリート・アクティビティにジョブの実行を指示する移動オブジェクト群本体と、を格納する記憶媒体が提供される。

【0018】本発明の他の一態様においては、ネットワーク上に分散する異なるプレースにおいてメッセージを送出する移動オブジェクト群の少なくとも一部を格納する記憶媒体であって、(a)第1のプレースで実行するジョブを定義する第1のコンクリート・アクティビティを保持する第1のキャリア・エージェントを特定する情報を含み、前記第1のプレースにおいて前記第1のキャリア・エージェントに対し、前記第1のコンクリート・アクティビティにジョブの実行を指示する第1のプロキシ・アクティビティと、(b)第2のプレースで実行するジョブを定義する第2のコンクリート・アクティビティを保持する第2のキャリア・エージェントを特定する情報を含み、前記第2のプレースにおいて前記第2のキャリア・エージェントに対し、前記第2のコンクリート・アクティビティにジョブの実行を指示する第2のプロキシ・アクティビティと、(c)前記第1のプロキシ・アクティビティと前記第2のプロキシ・アクティビティの実行順序を定義する順序リストと、を格納する記憶媒体が提供される。

【0019】本発明の他の一態様においては、ネットワーク上に分散するプレースにおいて、前記プレースに到

着したプロキシ・アクティビティを含む移動オブジェクト群本体の指示に応答してジョブを実行する移動オブジェクトを格納する記憶媒体であって、(a)前記プレースで実行するジョブを定義するコンクリート・アクティビティと、(b)前記プロキシ・アクティビティが前記コンクリート・アクティビティを特定するために前記プロキシ・アクティビティに提供されるコンクリート・アクティビティ・テーブルと、(c)前記プロキシ・アクティビティからジョブの実行命令メッセージを受領するためのメソッドと、を格納する記憶媒体が提供される。

【0020】本発明の他の一態様においては、ネットワーク上に分散するプレースにおいて、前記プレースに到着したプロキシ・アクティビティを含む移動オブジェクト群本体の指示に応答してジョブを実行する移動オブジェクトを格納する記憶媒体であって、(a)前記プレースで実行するジョブを定義するコンクリート・アクティビティと、(b)前記プロキシ・アクティビティからジョブの実行命令メッセージを受領するためのメソッドと、を格納する記憶媒体が提供される。

【0021】

【実施例】A. 概要

以下、図面を参照して本発明の実施例を説明する。図1を参照すると、本発明によって生成された移動オブジェクトを実行する分散ネットワーク環境150が示されている。各サーバ112～117には、移動エージェント125、135、141等にサービスを与えることが可能なプレース102～107が用意されている。この分散ネットワーク環境におけるプレース群はクラウドと呼ばれている。

【0022】クライアント・システム101においては、移動エージェントを生成するための移動エージェント生成部113が存在する。移動エージェント生成部113は、モデレータ・エージェント111に、生成したプランを渡し、モデレータ・エージェントはそのプランを実行することにより、指定されたプレースを移動していくことができる。

【0023】移動エージェント125等は各プレース102～107において存在する他のエージェントとコンタクトし(移動エージェントとコンタクトし、サービスを提供するエージェントを特にアクター・エージェントと呼ぶ)、リクエストを送出し、そのリクエストの結果を受けることができる。プレースは、エージェント間のコンタクトをサポートしている。また、移動エージェント125等は、アクター・エージェントから受領したリクエストの結果をResult139、143、153、163として保持し、移動を継続することができ、移動途中で、このResultを合成、分割等さまざまな加工を施すことも可能である。

【0024】本発明において、移動エージェントの移

動、分裂、消滅は、モデレータ・テンプレート181～187によって制御されており、ユーザは、その処理の態様により、さまざまなテンプレートを組合せ、図2に示すような分割や融合を含む複雑な作業を簡単に定義することができる。

【0025】B. ハードウェア構成

図3は、本発明の好適な実施例における移動エージェント生成システムのハードウェア構成の概観図が示されている。移動エージェント生成システム100は、中央処理装置(CPU)1とメモリ4とを含んでいる。CPU1とメモリ4は、バス2等を介して、補助記憶装置としてのハードディスク装置13、31とを接続してある。フロッピーディスク装置(またはMO、CD-ROM等の媒体駆動装置13、26、28、29、30)20はフロッピーディスクコントローラ(またはIDEコントローラ25、SCSIコントローラ27等の各種コントローラ)19を介してバス2へ接続されている。

【0026】フロッピーディスク装置(またはMO、CD-ROM等の媒体駆動装置26、28、29)20には、フロッピーディスク(またはMO、CD-ROM等の媒体)が挿入され、このフロッピーディスク等やハードディスク装置13、30、ROM14等の記憶媒体には、オペレーティングシステムと協働してCPU等に命令を与え、本発明を実施するためのコンピュータ・プログラムのコードを記録することができ、メモリ4にロードされることによって実行される。このコンピュータ・プログラムのコードは圧縮し、または、複数の分割して、複数の媒体にまたがって記録することもできる。

【0027】移動エージェント生成システム100は更に、ユーザ・インターフェース・ハードウェアを備えたシステムとすることができ、ユーザ・インターフェース・ハードウェアとしては、例えば、画面位置情報を入力をするためのポインティング・デバイス(マウス、ジョイスティック、トラックボール等)7、キー入力をサポートするキーボード6や、イメージデータをユーザに提示するためのディスプレイ11、12がある。また、スピーカ23は、オーディオ・コントローラ21から音声信号を、アンプ22を介して受領し、音声として出力する。

【0028】本発明の移動エージェント生成システム100の入力となるGUIプラン・ノード・ライブラリ(後に詳述する)は、ハードディスク30に格納されており、SCSIインターフェース27を介して本データベース検索システム100に入力される。シリアルポート15およびモデムまたは、トークンリング等の通信アダプタ18等を介して他のコンピュータ等と通信を行い、他のシステムのデータベースにアクセスすることや、フロッピーディスク24等の記憶媒体に存在するデータベースにアクセスすることも可能である。

【0029】このように、本発明は、通常のパーソナル

コンピュータ(PC)やワークステーション、テレビやFAX等の各種家電製品に組み込まれたコンピュータ又はこれらの組合せによって実施可能であることを容易に理解できるであろう。ただし、これらの構成要素は例示であり、その全ての構成要素が本発明の必須の構成要素となるわけではない。特に、本発明は、移動エージェントの生成を行うものであるため、シリアルポート15、通信アダプタカード18、オーディオ・コントローラ21、アンプ22、スピーカ23等の構成要素は本発明の一態様においては必須のものではない。

【0030】オペレーティング・システムとしては、Windows(マイクロソフト社の商標)、OS/2(IBM社の商標)、AIX(IBM社の商標)上のX-WINDOWシステム(MIT社の商標)などの、標準でGUIマルチウインドウ環境をサポートするものが望ましいが、特定のオペレーティング・システム環境に限定されるものではない。

【0031】また、図3は、スタンド・アロン環境のシステムを示しているが、クライアント/サーバ・システムとして本発明を実現し、クライアント・マシンは、サーバ・マシンに対して、イーサネット、トークン・リングなどでLAN接続し、サーバ・マシン側には、後述するプラン分割部等を配置し、その他の機能をクライアント・マシン側に配置してもよい。このように、サーバ・マシン側とクライアント・マシン側にどのような機能を配置するかは設計に際し自由に変更できる事項であり、複数のマシンを組合せ、それらにどのような機能を配分し、実施するか等の種々の変更は本発明の思想に包含される概念である。

【0032】C. システム構成

次に、図4、5のブロック図を参照して、本発明の好適な実施例における移動エージェント生成システムのシステム構成について説明する。

【0033】図4は、移動エージェント生成システム210における処理要素の一実施例を示す機能ブロック図である。実行コード生成部241は、プラン生成部213と、プランID生成部215を含んでいる。プランID生成部215は、プランにネットワーク上ユニークなプランIDを割りふる。本発明の好適な実施例においてプランID生成部215は、インターネットのURL、ポート番号、エージェントID(移動エージェント生成部に割りふられている)、シリアルNo. (移動エージェント生成部がプランID作成毎割りふる)を合成してプランIDを作成している。

【0034】プラン生成部213は、GUIプラン定義データ209に基づいてモデレータ・エージェントや、プランを生成する。作成されたプランには、実行するモデレータ・テンプレートやアクティビティの情報等を管理するプラン構成部223と、モデレータ・テンプレート、アクティビティの順序を定義する順序リスト225

を含んでいる。プラン構成部223は、プランノード・オブジェクト(モデレータ・テンプレート、アクティビティの総称)の関係付けの情報を有しており、ネットワークボロジ231を形成している。

【0035】GUIプラン定義データ生成部207は、オペレータの指定に基づいて、GUIプランノード・ライブラリ201の情報を使用してGUIプラン定義データ209を生成する。

【0036】プラン分割部は、実行コード生成部211によって生成されたプランから移動先プレース毎のキャリア・エージェント230を生成する。また、順序リストに対応付けられたアクティビティ205(後述するプロキシ・アクティビティと区別するためコンクリート・アクティビティと呼ぶ)をプロキシ・アクティビティ222に置換する。

【0037】プロキシ・アクティビティ222はコンクリート・アクティビティ205を実行させるためのメッセージを含み、本発明の好適な実施例においては実質的なジョブを実行するためのコードを含んではいない。キャリア・エージェント230は、各移動先プレースに転送され、移動オブジェクト群本体221の到着を待機し(逆に移動オブジェクト群本体221がキャリア・エージェント230の到着を待機する場合もある)、移動オブジェクト群本体221の指示に回答してコンクリート・アクティビティ205を実行する。

【0038】図5は、移動エージェント生成システム210における処理要素の一実施例をGUIプラン定義データ生成部207を中心に見た機能ブロック図である。GUIプラン定義データ生成部207へ入力情報を提供する入力イベント取得部231は、ユーザーからのキー入力、マウスのクリック等のイベントを取得し、GUIプラン定義データ生成部207のプラン・オブジェクト操作部233に解釈可能なメッセージに変換して渡す。

【0039】プラン・オブジェクト操作部231は、イベントの種類により、GUIアクティビティ、GUIテンプレート等のオブジェクトの生成や各種オブジェクトのプロパティの設定等の作業を行う。

【0040】プラン・オブジェクト属性変更部237は、指定されたオブジェクトのプロパティを変更するためのダイアログ・ウィンドウを提供する。修正された情報はプラン・オブジェクト操作部231を介してGUIプラン定義データ239に反映される。

【0041】GUIプラン定義データ保持部243は、プランを構成するアクティビティ、テンプレートの論理情報、図形情報を保持する。

【0042】プラン表示部245は、GUIプラン定義データ243の図形情報に基づき、画面上にプランの構造を視覚的に表示する。

【0043】実行コード生成部241は、GUIプラン定義データの論理情報に基づき、定義されたプランを生

成するための実行コード249(プラン)を生成する。

【0044】図6は、移動エージェント生成システム210によって生成された移動エージェント群270、280の実行時における処理要素の一実施例を示す機能ブロック図である。移動オブジェクト群本体270はモデレータ・エージェント251、プラン・オブジェクト261、モデレータ・テンプレート271、プロキシ・アクティビティ281を含んでいる。モデレータ・エージェント251は、移動エージェント・ライフサイクル制御部253とプラン実行部255を含んでいる。移動エージェント・ライフサイクル制御部253は、移動オブジェクト群本体270の移動、複製、消去等を実行する。プラン実行部255は、新たなプレースに到着したことを認識し、プラン・オブジェクト261にプランの実行を命令する。

【0045】この一方、プラン・オブジェクト261は、プラン実行制御部263、カレント・ノード267、順序リスト265を含んでいる。プラン実行制御部263は、モデレータ・エージェントのプラン実行部からプランの実行のインストラクションを受領すると、カレント・ノード267、順序リスト265を参照し、現在実行すべきプランノード・オブジェクト271、281を決定し、プランノード・オブジェクト271、281に実行を要求する。

【0046】カレント・ノード267は、後述するプランノードイタレータに対応するものであり、現在どのプランノード・オブジェクトを実行しているかを管理している。順序リスト265は、実行すべきプランノード・オブジェクト271、281を特定するリストである。分析とうが発生しない場合、プランノード・オブジェクト271、281は、この順序リスト265の順番にしたがってプランノードに実行を要求していく。

【0047】モデレータ・テンプレート271は、モデレータ・エージェント制御部273、プランステップ実行部275、アドレス参照部277を含んでいる。モデレータ・エージェント制御部273は、モデレータ・エージェント251の移動エージェント・ライフサイクル制御部253に対し、移動オブジェクト群本体270の移動、複製、消去を要求する。プラン・ステップ実行部275は、モデレータ・テンプレートの種類によって異なる定義となっており、移動前、移動後の夫々のプレースにおいて実行すべき命令等を管理している。アドレス参照部277は、プロキシ・アクティビティ281からアドレスを取得し、モデレータ・エージェント制御部273に移動すべきアドレスを教える等の役割を担っている。

【0048】プロキシ・アクティビティ281は、プロキシ・アクティビティ実行部283とメッセージ機能285を含んでいる。プロキシ・アクティビティ実行部283は、メッセージ機能285を介してキャリア

・エージェントと対話し、そのプロキシ・アクティビティに対応するコンクリート・アクティビティ282を有するキャリア・エージェント280か否かを判断し、対応するコンクリート・アクティビティ282に対しジョブの実行を指示する。コンクリート・アクティビティ・テーブル269は、実行すべきコンクリート・アクティビティ282を特定するためのテーブルである。

【0049】コンクリート・アクティビティ282は、アクティビティ実行部284とメッセージ機能286を含んでいる。アクティビティ実行部284は、移動エージェントが渡り歩く各プレースで実行するインストラクションや各プレースで受け取ったResultを格納する機能を有している。メッセージ機能286は、アクター・エージェント291とのメッセージをやり取りする機能を有している。

【0050】アクター・エージェント291は、プレース上に存在する他のエージェントであり、コンクリート・アクティビティ282のリクエストに対し、サービスを提供する機能を有している。このアクター・エージェント291も、アクティビティとメッセージのやり取りするメッセージ機能293を有し、また、リクエストの内容を判断し、サービスを提供するメッセージ処理部295を有している。

【0051】以上図4乃至6に示す各機能ブロックを説明したが、これらの機能ブロックは、論理的機能ブロックであり、各々1つのまとまりを持ったハードウェアやソフトウェアによって実現されることを意味するのではなく、複合し、または共通したハードウェアやソフトウェアによって実現可能である。また、この図4乃至6に示す全ての機能ブロックが本発明に必須の構成要素という訳ではない。例えば、順序付リスト265は、カレント・ノード管理部267が直接カレント・ノードを管理し、また、ANDJoin等の処理においても、他のResultを取得するモデレータ・エージェントを決めておけばよいので、特に必要な構成要素ではない。さらに、実行コード生成部を他のマシン上に配置し、そのマ

シンにGUIプラン定義データを送信することによっても実行コードを生成することができる。

【0052】D. オブジェクト構成

D-1. 移動エージェント群のオブジェクト構成

図7、8は、本発明の移動オブジェクト群270、280のオブジェクト図である。モデレータ・エージェント303、プラン305、プランID、プラン・イタレータ307の各クラスは、メソッドを有しており、プランノード311、アクティビティ313、モデレータ・テンプレート315、及びChain等のテンプレート321〜326の各クラスは、データ及びメソッドを有している。

【0053】この図において、三角印302、314、316は、上位のクラスのデータやメソッドを承継していることを示し、黒丸312はその下位のクラスが複数存在しえることを示している。プランノードは抽象的なクラスで、モデレータ・テンプレートとアクティビティクラスの双方を共通のインターフェースで定義する。これによってプランは、プランの生成や実行中に双方のクラスを同等に扱うことができる。

【0054】移動エージェント301は分散コンピュータ環境上の各プレースで現在提供されている機能（プレース・クラス）であり、このクラスに移動エージェントの作成を指示するAPIを投げることによって、ユーザは簡単に移動エージェント（モデレータ・エージェント）を作成することができる。現在、従来技術の説明で上述したJavaベースのモバイル・エージェントやテレスクリプトの移動エージェントの双方でサポートされており機能である。

【0055】本発明の好適な実施例においては、コンクリート・アクティビティとプロキシ・アクティビティを置換すべく、また、キャリア・エージェントを生成すべく、以下のメソッドが提供されている。

【0056】

【表1】

```
setProxyActivity (index, agentid) {
    //アクティビティの入れ替え
    conAct = -plan.elementAt(index);
    address = conAct.getdestination();
    proxyAct = new ProxyActivity (address, agentid);
    -plan.setPlanNode(proxyAct, index);
    //キャリア・エージェントの生成
    carrierAgent = context.getAglet(agentid)
    carrierAgent.sendMessage("store", conAct, index);
}
```

【0057】プランノード・クラスは、サクセッサとプレデセッサのリストを有しており、複数のプランノードの関係を管理している。プランノードのクラスは"setPredecessor""setSuccessor"のメソッドを提供し、ノード

間の関係を定義している。プランノードの具体的なクラスは、これらのメソッド間の関係を見分けることができる。

【0058】モデレータ・テンプレートは、異なるプレ

ースで実行されるアクティビティと連結するため、オリジン（移動前のプレース）かディステーション（移動後のプレース）かを区別するステート情報を有している。本発明の好適な実施例においては、モデレータ・テンプレートはアクティビティ・オブジェクトとだけ結合することができる。モデレータ・テンプレートにはChainを始め、複数のテンプレートが提供されており、複数の分裂したモデレータ・エージェントが分担して仕事をするための基本的メカニズム等の提供を可能にしている。

【0059】アクティビティは、ジョブを実行すべきプレースのアドレス情報を保持している。また、「abstractActivity」を公開することにより、以下に示す「executeActivity」等の外部からの指示に応答して、内部に保有するジョブを実行することができる。プロキシ・アクティビティは、対応するコンクリート・アクティビティを保持するキャリア・エージェントを特定するキャリア・エージェントIDの情報を有し、コンクリート・アクティビティに対してジョブの実行を指示する。

【0060】

【表2】

```
executeActivity {
    index = -plan.indexOf(this)
    agent = context.getAgent(id)
    table = agent.sendMessage("gettable")
    conAct = table.elementAt(index);
    object result = conAct.executeActivity ();
}
```

【0061】D-2. GUIプラン定義データのオブジェクト構成

図9は、本発明の好適な実施例におけるGUIプラン定義データ239のオブジェクト図である。GUIアクティビティ403、GUIモデレータ・テンプレート413、GUIプレース405及びリンク407等の各クラスは、データ及びメソッドを有している。

【0062】この図において、三角印302、314、316は、上位のクラスのデータやメソッドを承継していることを示し、黒丸312はその下位のクラスが複数存在しえることを示している。

【0063】GUIアクティビティ403は、GUIビルダ上でアクティビティとして表示されるためのオブジェクトであり、ディスプレイ上に表示されるときに座標値（表示開始位置：x1、y1と表示終了位置：x2、y2）とリンクするオブジェクトを特定するサクセッサ・リスト及びプレデセッサ・リストのデータを有している。

【0064】GUIモデレータ・テンプレート409もこれと同様に、GUIビルダ上でモデレータ・テンプレートとして表示されるためのオブジェクトであり、ディスプレイ上に表示されるときに座標値（表示開始位置：

x1、y1と表示終了位置：x2、y2）とリンクするオブジェクトを特定するサクセッサ・リスト及びプレデセッサ・リストのデータを有している。また、モデレータ・テンプレートの種類を特定するためのテンプレートタイプのデータも有している。

【0065】リンク・オブジェクト407は、GUIアクティビティ403とGUIモデレータ・テンプレート409間のリンク関係を表示するためのオブジェクトであり、ディスプレイ上に表示されるときに座標値（表示開始位置：x1、y1と表示終了位置：x2、y2）とリンクするオブジェクトを特定するサクセッサ・リスト及びプレデセッサ・リストのデータを有している。

【0066】GUIプレース405は、各アクティビティが実行されるプレースを表示するためのオブジェクトであり、包括して表示すべきGUIアクティビティの座標値情報（各GUIアクティビティの表示開始位置：x1、y1と表示終了位置：x2、y2）とプレースのアドレス情報のデータを有している。

【0067】D-3. 移動エージェント群のメッセージ・フロー

図10は、本発明の好適な実施例における各クラスの主要なメッセージフローを示す図である。モデレータ・エージェント351はプラン353に対しnextStep（次のステップを実行することを指示するインストラクション）を送出し、プランはこれに回答して、プランノード・オブジェクト（モデレータ・テンプレート357またはプロキシ・アクティビティ355）に対しnextStepを送出する。モデレータ・テンプレート357は、その種類や状況に応じて、プロキシ・アクティビティ355にgetARL（アドレスの送出命令）を送出や、isMarked（マーク付けをする）を送出する。また、モデレータ・エージェント351に対しclone（複製の作成要求）、dispatch（移動要求）、dispose（消滅要求）を送出する。

【0068】プロキシ・アクティビティ355は、キャリア・エージェント352に対し、getTableを送出し、コンクリート・アクティビティ・テーブル354を入手する。そして、コンクリート・アクティビティ・テーブル354に対して自己の保有するインデックス（順序リストのインデックス）をキーに対応するコンクリート・アクティビティ358を入手する。そして、プロキシ・アクティビティ355は、コンクリート・アクティビティ358にユーザの指定したリクエストを実行させるため、コンクリート・アクティビティ358に対しexecuteActivity（アクティビティの実行要求）を送出する。

【0069】これに回答して、コンクリート・アクティビティ358は、ユーザの指定したリクエストを実行するため、アクター・エージェント359に対しsend

Message (メッセージの送付要求)を送出する。

【0070】E. プラン定義データの作成

本発明においては、この移動エージェントの生成をユーザが簡単に行うことを可能にする開発支援ツールを提供している。図11乃至図16は、本発明の好適な実施例における開発支援ツールのユーザインターフェースを示す図である。ユーザはこの開発支援ツールを使用することによって、上記プラン定義部のコーディングを簡単に生成することができる。

【0071】まず、ユーザがプルダウン・メニューから作成371を選択し、ポインティング・デバイスでクリックすると、プラン・オブジェクト操作部233は、プラン名を入力するためのダイアログボックス375を表示する。これにより、ユーザはプランIDとは異なるユーザにとって親しみやすい名前でプランを管理することができる。次に、ユーザがプラン名をプラン名の入力エントリ376入力し、OKボタン377を押すと、プラン定義用のウィンドウ380が開かれる。プラン名を入力するためのダイアログボックス375は、プルダウン・メニューの保存372や終了374を選択時に表示してもよい。

【0072】ユーザがポインティング・デバイスでアクティビティ・ボックス420のアクティビティ・アイコン(ACT1)421をプラン定義用ウィンドウ380にドラッグしドロップする。すると、図9に示すGUIアクティビティ403及びGUIプレース405の各オブジェクトがGUIプランノード・ライブラリ201のひな型のデータを基に生成される。生成されたGUIアクティビティ403のDisplayメソッドが呼ばれ、図12に示すようにプラン定義用ウィンドウ380にGUIアクティビティとして表示される。このとき、GUIアクティビティ403のプレデセッサ・リストとサクセッサ・リストには「リンクするオブジェクトなし」を示す「0」がセットされる。

【0073】このGUIアクティビティ431は、図9に示すようにSelectメソッドやMoveメソッドを有しており、マウスポインタによるセレクトに対応した処理の実行や、マウสดラッグに対して表示位置を変更することができる。

【0074】次に、ユーザがポインティング・デバイスでテンプレート・ボックス390のAND-Splitアイコン392をプラン定義用ウィンドウ380にドラッグしドロップする。すると、図9に示すGUIモデレータ・テンプレート410及びリンク407の各オブジェクト(リンク・オブジェクトは2つ生成される)がGUIプランノード・ライブラリ201のひな型のデータを基に生成される。生成されたGUIモデレータ・テンプレート413のDisplayメソッド、リンク・オブジェクト407のDisplayメソッドが呼ばれ、図13に示すようにプラン定義用ウィンドウ380にG

UIモデレータ・テンプレート433およびリンク・オブジェクト432、434として表示される。

【0075】このとき、リンク・オブジェクト432のサクセッサ・リストにはGUIモデレータ・テンプレート433の情報がセットされ、リンク・オブジェクト434のプレデセッサ・リストにはGUIモデレータ・テンプレート433の情報がセットされる。この一方、GUIモデレータ・テンプレート433のプレデセッサ・リスト及びサクセッサ・リストには、それぞれリンク・オブジェクト432、434を特定する情報がセットされる。各リンク・オブジェクトの表示位置は、GUIモデレータ・テンプレート433の位置情報から算出された値がセットされる。

【0076】このGUIモデレータ・テンプレート433も、図9に示すようにSelectメソッドやMoveメソッドを有しており、マウスポインタによるセレクトに対応した処理の実行や、マウสดラッグに対して表示位置を変更することができる。GUIモデレータ・テンプレート433の移動がなされると、GUIモデレータ・テンプレート433のプレデセッサ・リスト及びサクセッサ・リストで指定されたオブジェクトや、そのオブジェクトにリンクする一連のオブジェクトの表示位置を移動量に応じて更新し、Displayメソッドによる最描画をおこなう。

【0077】その後、図14に示すように、ユーザがリンク・オブジェクト432をGUIアクティビティ431上にドラッグ&ドロップすると、プラン・オブジェクト操作部233は、GUIアクティビティの表示位置とドラッグ中のマウスポインタの位置を比較することにより、リンクするGUIアクティビティを特定し、GUIアクティビティ431のサクセッサ・リストにリンク・オブジェクト432を特定する情報をセットし、リンク・オブジェクト432のプレデセッサ・リストにGUIアクティビティ431を特定する情報をセットする。また、リンク・オブジェクト432の表示位置をGUIアクティビティ431の表示位置に応じた値に更新し、Displayメソッドによる最描画をおこなう。

【0078】このようにして、ユーザは、移動エージェントに行わせる仕事の性質や手順によって、図15に示すように、所望の実行順序のアクティビティを配置することができる。その後、ユーザが同一のプレースにおいて実行を希望するアクティビティを、キーボードのコントロールキーをおしながら選択し、プレースのアイコン399をクリックすると、図16に示すように、選択したアクティビティを包括する矩形501~503が表示され、プレース・プロパティ設定ウィンドウ510が表示される。

【0079】本発明の好適な実施例においては、選択されたアクティビティの中心座標をx軸方向に順番に並べ、隣り合うアクティビティの中心座標を結ぶ線分を中

心とした一定幅の長方形を算出することにより表示すべき矩形501〜503を求めている。なお、これらの矩形はアクティビティの中心座標を焦点とする楕円などによっても代替可能である。

【0080】なお、本発明の好適な実施例において、3つ以上のアクティビティが同一のプレースと指定された場合、矩形の重なり合った部分は除去され、1つの多角形として表示される。また、GUIプレースを象徴する各々の矩形または多角形は、異なる色彩アトリビュートが割り当てられ、オペレータはプレースの違いを容易に判断することができる。

【0081】この例においては、アクティビティを、キーボードのコントロールキーをおしながら選択し、プレースのアイコン399をクリックすることにより、同一のプレースにおいて実行を希望するアクティビティを選択しているが、マウスのポイント&ドラッグにより、同一のプレースにおいて実行を希望するアクティビティを囲い込むオペレーションや選択後のクリックによる選択取り消し等のオペレーションによっても代替可能なものである。

【0082】ユーザがプレース名やプレースアドレスをセットすると、選択されたGUIアクティビティ403（図9）のGUIプレース405のアドレス、プレース名として設定される。

```
public class Planner extends StationaryAgent {
    private Plan _plan;
    private PlannerFrame f;
    private URL _home;
    private AgletIdentifier _id;
    private int _planIndex = 0;
    public void onCreate(Object args) {
        //プランの選択のためウインドウを生成する。
        f = new PlannerFrame("JMT - PlannerAgent", this);
        f.show();
        try {
            //実行コード生成部の存在しているURLを入手
            _home = getAgletContext().getHostingURL();
            //移動エージェント生成部のIDを入手
            _id = getIdentifier();
        } catch (InvalidAgletException e) {
            Utility.print(this, e.getMessage());
        }
    }
}
```

【0087】本発明の好適な実施例においては、プラン・オブジェクトの生成を行うための「New演算子」がJavaの機能として提供されており、プランIDを引数にしてこのクラスを実行することにより、新たなプラン・オブジェクトが生成される。

【0088】なお、モデレータ・エージェントは、上述のように、各プレースで現在提供されている機能（プレ

【0083】F. プランの生成

図17は、プラン定義データを基に実行コード（プラン）を生成する手順を示すフローチャートである。まず、プラン・オブジェクトを生成する（ブロック603）。プランの生成においては、大きく分けて、プランIDの生成と、プラン・オブジェクトの下位に位置するオブジェクトとのリンク状態を確立する工程に分けられる。

【0084】本発明の好適な実施例においては、プラン・オブジェクトにグローバルなネットワーク上で一意に決まるユニークなプランIDを割りふられる。本発明の好適な実施例においてプランID生成部215（図4）は、インターネットのURL、ポート番号、エージェントID（移動エージェント生成部に割りふられている）、シリアルNo.（移動エージェント生成部がプランID作成毎割りふる）を合成してプランIDを作成している。

【0085】このプランID生成部のコーディングの例を表1に示す。本発明の好適な実施例においては、プランID生成部のインストラクション群はプログラム部品としてユーザに提供されている。

【0086】

【表3】

ース・クラス）を利用して生成される。本発明の好適な実施例においては、Javaベースのモバイル・エージェントのAgletsクラスに移動エージェントの作成を指示するAPIを投げることによって、モデレータ・エージェントの生成を行っている。

【0089】次に、先頭のGUIアクティビティに対応するアクティビティを生成する（ブロック605）。具

体的には、まず、GUIアクティビティのアクティビティ・タイプから実行時のクラス名（この場合はA1）を取得し、その後、クラス名「A1」をキーにそのインスタンスを生成する。生成されたアクティビティはプラン・オブジェクトに追加される（ブロック607）。

【0090】仕掛かり中のGUIオブジェクト（GUIアクティビティまたはGUIモデレータ・テンプレートまたはリンク・オブジェクト）を格納するFIFO（First In First Out）キューを作成し（ブロック609）、先頭のGUIオブジェクト（GUIアクティビティ）をキューに入れる（ブロック611）。先頭のGUIオブジェクトであるか否かは、プレデセッサ・リストがnullであるか否か等によって判別することができる。

【0091】次に、生成されたキューから要素（GUIオブジェクト）を取り出し（ブロック612）、図18に示すサクセッサ・リストにあるアクティビティまたはモデレータ・テンプレートを追加するサブルーチン（ブロック613）を実行する。

【0092】図18は、このサブルーチンの処理手順をしめすフローチャートである。まず、サクセッサ・リストの要素数と現在のiの値を比較し、全てのサクセッサ・リストの要素数について、この処理が実行されたか否かを判断する（ブロック665）。

【0093】サクセッサ・リストの要素数>iの場合は、サクセッサ・リストのi番目のGUIオブジェクトを取り出す（ブロック657）。

【0094】そして、取り出したGUIオブジェクトが、すでにキューに存在しているか否かを判断し（ブロック658）、存在していない要素のみを図17のブロック609で生成したキューに入れる（ブロック659）。

【0095】次に、取り出した要素（GUIオブジェクト）がGUIアクティビティまたはGUIモデレータ・テンプレートであるか否かを検査する（ブロック660）。取り出した要素がGUIアクティビティまたはGUIモデレータ・テンプレートである場合には、対応するアクティビティまたはモデレータ・テンプレートを作成する（ブロック661）。このとき、GUIアクティビティの場合には、対応するGUIプレースのアドレス情報を入手する。

【0096】そして、カレントのGUIオブジェクトのプレデセッサ・リストのリンク・オブジェクトよりプレデセッサ・インデックスを取得し、自己のプレデセッサ・リストにセットする。なお、最初でないGUIアクティビティまたはGUIモデレータ・テンプレートには、必ずプレデセッサ・リストにはリンク・オブジェクトの情報が入っている。最初のGUIアクティビティには、プレデセッサ・リストにnullの情報がセットされているので、自己のプレデセッサ・リストにもnullを

セットする。

【0097】次に作成した要素（アクティビティまたはモデレータ・テンプレート）をプランに追加し（ブロック663）、プラン中の順序リストのインデックスをカレントのGUIオブジェクトのサクセッサであるリンクオブジェクトにセットする。リンクオブジェクトはこの値をプレデセッサ・インデックスとして保持し、アクティビティまたはモデレータ・テンプレートのプレデセッサ・リストを形成するための情報として使用される。

【0098】取り出した要素がGUIアクティビティまたはGUIモデレータ・テンプレートでない場合、対応するアクティビティまたはモデレータ・テンプレートを作成する処理等は行わず、次の要素の処理をおこなう。

【0099】そして、このブロック657乃至ブロック665の処理をサクセッサ・リストの要素の数繰り返した後、再び図17に戻り、キューに要素はあるか否か判断し、まだある場合には要素を取り出し（ブロック612）、再び図18に示すサクセッサ・リストにあるアクティビティまたはモデレータ・テンプレートを追加するサブルーチン（ブロック613）を実行する。

【0100】G. キャリア・エージェントの生成とプロキシ・エージェントへの置換、各プレースへのキャリア・エージェントの送信

プランの生成が完了すると、図4に示したプラン分割部224が起動され、各移動先プレース毎のキャリア・エージェント230を生成し、順序リスト225に対応付けられたコンクリート・アクティビティ205をプロキシ・アクティビティ222に置換する処理を行う。また、生成されたキャリア・エージェント230は、各移動先プレースに転送される。

【0101】図19、20は、プラン分割部224と生成されたキャリア・エージェント230の行うキャリア・エージェントの生成とプロキシ・エージェントへの置換、各プレースへのキャリア・エージェントの送信の処理の手順を示すフローチャートである。

【0102】この処理が開始すると（ブロック731）、まずiが初期化され（ブロック732）、プランからi番目の要素が取り出される（ブロック733）。そして、この取り出された要素がアクティビティ（コンクリート・アクティビティ）であるか否か判断される（ブロック734）。取り出された要素がアクティビティでない場合には、次の要素を取り出す（ブロック735、733）。

【0103】本発明の好適な実施例においては、全てのコンクリート・アクティビティをプロキシ・アクティビティに置換し、コンクリート・アクティビティをキャリア・エージェントに乗せて各プレースに転送しているが、ブロック734を特定のコンクリート・アクティビティか否かという判断に変更することで、一部のコンクリート・アクティビティのみをプロキシ・アクティビ

ティに置換することもできる。

【0104】取り出された要素がアクティビティである場合には、プロキシ・アクティビティを生成し（ブロック736）、コンクリート・アクティビティから取り出した移動先プレースのアドレスをセットする（ブロック737）。そして次に、ワークテーブルを参照し、同じ移動先プレース・アドレスを有するキャリア・エージェントが存在するか否かを判断する。

【0105】同じ移動先プレース・アドレスを有するキャリア・エージェントが存在しない場合には、このコンクリート・アクティビティを運搬するためのキャリア・エージェントを生成し（ブロック739）、生成されたキャリア・エージェントにエージェントID及び移動先プレース・アドレスをセットする（ブロック740）。なお、本発明の好適な実施例において、このキャリア・エージェントIDは、プランIDとプレースに対応したシリアル番号によって生成されている。

【0106】そして、キャリア・エージェントのIDと移動先プレース・アドレスをワークテーブルにセットする。また、i番目のコンクリート・アクティビティをキャリア・エージェントのコンクリート・アクティビティ・テーブルのi番目にセットする（ブロック742）。キャリアエージェントには、タイム・ボムをセットし、所定の時期に消滅させることもできる。

【0107】なお、過去に有効な同一のコンクリート・アクティビティを保持し、同一プレースに送出されたキャリア・エージェントが存在する場合には（ワークテーブルを一定期間保持すること等により判別可能である）、キャリア・エージェントを生成せず、プロキシ・アクティビティに過去に送出したキャリア・エージェントの同一のコンクリート・アクティビティを特定して実行を指示するようエージェントIDとインデックス情報を置換することも可能である。

【0108】同じ移動先プレース・アドレスを有するキャリア・エージェントが存在する場合には、そのキャリア・エージェントのコンクリート・アクティビティ・テーブルのi番目にセットする（ブロック742）。

【0109】この一方、プロキシ・アクティビティにはキャリア・エージェントのIDをセットし、これをプランのi番目にセット（リプレース）する。このブロック733から744の処理を全てのプランの要素について行うことにより、各移動先プレースに対応したキャリア・エージェントの生成と移動オブジェクト群本体のコンクリート・アクティビティをプロキシ・エージェントに置換する作業が完了する（ブロック745、746）。

【0110】そして、生成したキャリア・エージェントを各移動先プレースへ転送する（ブロック748）。

【0111】以上説明した例は、移動オブジェクト群本体が直接コンクリート・アクティビティを保持して移動

していたものを改良して実施化したものであるため、一旦プランがコンクリート・アクティビティを保持するものを生成し、その後コンクリート・アクティビティをプロキシ・アクティビティに置換している。しかし、移動オブジェクト群本体の生成時にプランが直接プロキシ・アクティビティを保持するものを生成させることも可能である。かかる態様の実施も本発明の思想に包含されるものである。

【0112】H. プランの実行

H-1. Chainテンプレートを含むトポロジーの場合

【0113】Chainテンプレートを含むトポロジーの場合における移動オブジェクト群の動作を図21乃至図25を使用して説明する。図21は、プランの実行手順を示すフローチャートであり、図22は、モデレータ・テンプレートの機能ブロック図である。図25は、移動オブジェクト群がオリジンのプレースからディステーションのプレースに移動する状態の概念図である。まず、図21を参照する。モデレータ・エージェント351が生成され、プラン353が割り当てられると（ブロック843）、モデレータ・エージェントはプランを自動的に実行する（ブロック845）。

【0114】モデレータ・エージェント351からプラン・オブジェクト353に対し送出された、“NextStep”のインストラクションに応答して、プラン・オブジェクト353はまず、カレントのプランノードを取り出す（ブロック847）。本発明の好適な実施例においては、プラン・イタレータオブジェクト352が現在プランノードのどのステップにあるかを管理しており、プラン・オブジェクト353の問い合わせに回答して、現在プランノードのどのステップにいるか（カレント）の情報を提供する。

【0115】カレントの問い合わせに対し、プラン・イタレータオブジェクト352が「0」（最初のステップ）を返した場合、プラン・オブジェクト353は、act1を取り出す。そして、プラン・オブジェクト353は、act1に対して、その中で定義されているアクティビティにアクティビティが管理しているリクエストを実行することを依頼する（ブロック849）。

【0116】アクティビティは内部に保持しているエージェントのクラス名（エージェント名）が同一のプレースに存在しているか否かを判断する。図23に示すように、モデレータ・エージェント350はプレース360に到達すると、プレース360に対し自分のエージェント名を教え、プレース360はこれをエージェントリスト369として管理している。そして、プレース360に存在するエージェントは、プレース360に対し“get AgentList”のインストラクションを出し、問い合わせを行うと、プレースはAgentListを返す機能があるので、同一プレースに存在するエージェントを知ることができ

る。

【0117】`act1`はプロキシ・アクティビティに置換されているため、まず、自己の保有するキャリア・エージェントIDに対応するエージェントが存在するか否かを判断し、存在しない場合には、キャリア・エージェントの到着を待機する（スリープモードに入る）。

【0118】対応するエージェントが存在する場合には、そのキャリア・エージェントに対して、コンクリート・アクティビティ・リストの対応するインデックス（そのプロキシ・アクティビティが対応付けられている順序リストの番号）のアクティビティを実行することを指示する。

【0119】コンクリート・アクティビティは、これに応答して内部に保持しているメッセージの内容にしたがってアクター・エージェントを探索する。そして、アクティビティは見つけたエージェント（アクター・エージェント）361に対し内部に保持しているメッセージ（リクエスト）を送る。アクター・エージェント361はこのリクエストに対し、リターンオブジェクトをリクエスト側のコンクリート・アクティビティ358に送出する。さらに、コンクリート・アクティビティ358は、リクエスト側のプロキシ・アクティビティに、この受け取ったオブジェクト実行命令のリターンとして送出する。プロキシ・アクティビティは受領したオブジェクトを`Result`のレコード525に格納する。

【0120】プロキシ・アクティビティはプラン・オブジェクト353に依頼の仕事が終わったことを知らせ、これに回答してプラン・オブジェクト353は`act1`に処理が終了したことを示すマークを付ける（ブロック855）。そして、プラン・オブジェクト353はプラン・イタレータ352に対し、ステップを進めることを依頼し（ブロック859）、プラン・イタレータ352は、プラン・オブジェクト353にカレントの情報を提供する。

【0121】この場合、カレントは“1”（2番目のステップ）であるので、プラン・オブジェクトは`Chain`テンプレートのオブジェクトを取り出す（ブロック845、847）。そして、プラン・オブジェクトは`Chain`に対して、実行を依頼する（ブロック849）。

【0122】`Chain`のモデレーション管理マネージャ603は、ステート管理部601に現在のステートを問い合わせる。ステート管理部601は、デフォルトでオリジンの状態になっているので、オリジンであることをモデレーション管理マネージャ603に知らせる。モデレーション管理マネージャ603は、ステート管理部601から、ステートの情報を得ると、ステートの切り換えをステート管理部601に依頼する。ステート管理部601は、これに回答してステートをオリジンからディスティネーションに切り換える。

【0123】このテンプレートの行うステートの切り換

え及びオリジンまたはディスティネーションにおける処理のフローを図24に示す。このフローチャートは複数種類存在するテンプレートに共通な動作を示しており、ブロック8257のオリジンにおける処理とブロック831のディスティネーションにおける処理の内容がテンプレートの種類によって異なる処理が実施される。

【0124】モデレーション管理マネージャ603は、オリジン用モデレーション・モジュール605に処理を依頼する。オリジン用モデレーション・モジュール605とディスティネーション用モデレーション・モジュール607には、夫々のプレースにおいてテンプレートが動作するためのインストラクションのセットが管理されている。

【0125】オリジン用モデレーション・モジュール605は、モデレータ・エージェント参照モジュール599に問い合わせを行うことにより、現在モデレータ・エージェント351の存在するプレースのアドレスを取得する。

【0126】また、サクセッサ・リスト609に対し、サクセッサ・リスト609に登録されているアクティビティのアドレスの渡すように依頼する。サクセッサ・リスト609は、登録されているアクティビティ（`act2`）からアドレスを取得し、オリジン用モデレーション・モジュール603に渡す。

【0127】オリジン用モデレーション・モジュール603は、アクティビティのアドレスと現在モデレータ・エージェントの存在するプレースのアドレスを比較し、もし2つが異なっていたらアクティビティのアドレスにモデレータ・エージェントを移動させる。具体的には、オリジン用モデレーション・モジュール605は、モデレータ・エージェント参照モジュールを介して、移動を命令するインストラクションを送出する。同一アドレスの場合には、処理を終了する。

【0128】プラン・オブジェクトは移動先で再びプラン・オブジェクトに対し1ステップの実行を依頼する。同様にカレントのプランノードを取り出す。プラン・オブジェクトは`Chain`に対し、次のステップの実行を依頼する。`Chain`はステートの情報を確認し、ディスティネーションの処理を行う。ステートの情報をオリジンの状態に戻す。

【0129】プランは`Chain`のディスティネーションの処理の終了を確認すると、次のカレントのプランノードを取り出し（`act2`）、アクティビティの実行を依頼する。`act2`は、`act1`が内部に保持する`result`を取得する。

【0130】本発明の好適な実施例においては、ディスティネーションモジュール607がブレードセッサ・リスト591を参照し、`Chain`に対し、処理結果の送出を依頼する“`getResult`”のインストラクションを送出する。`Chain`は、これに回答し、自己の保有する`Re`

su1tを探索する。しかし、ChainはResultを保有していないので、さらにプレデッサ・リスト591を参照し、プレデッサ・リスト591に登録されたオブジェクト(act1)にResultをもらいに行き、そのResultをact2に返す。プレデッサ・リスト591に登録されたオブジェクトが存在しない場合には、act2にResultを保有していないことを示すDummyを送出する。この処理はテンプレートにResultを格納するレコードを作成することによって代替可能である。

【0131】act2は、内部に保持しているエージェントのクラス名をキーに同一スペースに存在するレポートエージェントの参照を得る。act2はレポートエージェントに対し内部に保持しているメッセージをact1から入手したresultを引き数にして送る。プランは次のカレント・ノードを取り出す。この場合、カレント・ノードは存在しないので、プランはこれを検出して処理を終了する。ディスティネーション用モデレーション・モジュール607はモデレータ・エージェント参照モジュール599を介して、モデレータ・エージェントにモデレータ・エージェントを消滅させるインストラクションを送出する。

【0132】H-2. Iterationテンプレートを含むトポロジーの場合

図26は、Iterationテンプレートによって制御される移動オブジェクト群の動作の概要を示す図である。以下、図22を参照し、Iterationテンプレートを含むトポロジーの場合の処理を説明する。Chainの場合と同様に、モデレータ・エージェント351を生成し、その際の引き数として作成したプラン・オブジェクト353を渡し、プラン353を1ステップずつ実行する。また、プラン・オブジェクト353は、Chainの場合と同様に、カレントのプランノード(Chain)を取り出し、Chainはオリジンにおける処理を実行する。また、移動先において、「H-1. Chainテンプレートを含むトポロジーの場合」で説明したように、ディスティネーションにおける処理の実行、act1のアクティビティの実行を行う。

【0133】act1の処理が終了すると、次にプラン353はIterationに次のステップの実行を依頼する。図27は、Iterationテンプレートのオリジンにおける処理手順を示すフローチャートである。Iterationテンプレートのオリジンにおける処理においては、まず、次のステップに進んでよいのか、例えば、act1のresultの情報が所定の条件に一致しているか否かチェックを行う(ブロック873)。

【0134】もし条件を満たしている場合、内部に保持するサクセッサ情報をサクセッサ・リストに格納する(ブロック875)。そして、サクセッサ・リストのa

ct2よりアドレスを取り出し(ブロック877)、そのアドレスにモデレータ・エージェントを移動させる(ブロック879)。もし、所定の条件に一致していない場合には、ステートをオリジンにし(ブロック881)、リターン・ノード・インデックスよりプランノードの参照を得、そのプランノードでサクセッサ・リストの要素を置き換え、次のステップに進む(ブロック883)。以下「H-1. Chainテンプレートを含むトポロジーの場合」において説明したようにact2の処理を行い、処理を終了する。

【0135】H-3. ANDSplit及びANDJoinテンプレートを含むトポロジーの場合

図28、29は、ANDSplit、ANDJoinテンプレートによって制御される移動オブジェクト群の動作の概要を示す図である。以下、図22を参照し、ANDSplit及びANDJoinテンプレートを含むトポロジーの場合の処理を説明する。Chainの場合と同様に、モデレータ・エージェント351を生成し、その際の引き数として作成したプラン・オブジェクト353を渡し、プラン353は、1ステップずつ実行される。

【0136】また、プラン・オブジェクト353は、Chainの場合と同様に、カレントのプランノードであるANDSplitを取り出す。プラン353はANDSplitに対して次のステップの実行を依頼する。図30は、ANDSplitテンプレートにおけるオリジンの処理の処理手順を示すフローチャートである。

【0137】モデレーション管理マネージャ603は、ステート管理部601のステート情報を検索し、デフォルトである、「オリジン」の情報を取得する。モデレーション管理マネージャ603は、オリジン用モデレーション・モジュール605にオリジンにおける処理を行うよう依頼する。モデレーション管理マネージャ603は、ステートをディスティネーションに変えるようステート管理部に依頼する(ブロック893)。そして、ANDSplitは、サクセッサ・リストを参照し(ブロック897)、サクセッサ・リストの最初のアクティビティを取り出し、マークド・インデックス595に「0」をセットする(ブロック899)。

【0138】次に、モデレーション管理マネージャ603は、モデレータ・エージェントのクローンを作成するようモデレータ・エージェント参照モジュール599を介してモデレータ・エージェント351に依頼する(ブロック901)。モデレータ・エージェントは、この依頼に回答して、モデレータ・エージェント及びモデレータ・エージェントの保持するプラン、テンプレート、アクティビティ等、全てのオブジェクトのコピーも作成する。

【0139】コピーによって作成されたクローンのモデレータ・エージェントは、クローンの作成の完了に回答して、自分のエージェントIDをモデレータ・エージェ

ント参照モジュール599に教える。モデレータ・エージェント参照モジュール599は、クローンの作成が完了したことをオリジン用モジュレーションモジュール605に知らせる。

【0140】オリジン用モジュレーションモジュール605はこれに応答して、マークド・インデックスの値(0)に対応するアクティビティ(act1)のアドレスをサクセッサ・リスト609から取得する。オリジン用モデレータモジュール605は、モデレータ・エージェント参照モジュール599を介して、クローンをact1のアドレスに対応するプレースに移動することをモデレータ・エージェント351に依頼する(ブロック903)。

【0141】オリジン用モジュレーションモジュール605は、サクセッサ・リスト609を参照し、次のアクティビティが登録されているか否かを判断し、登録されている場合には、さらに、もう一つのクローンの作成の処理を同様の手順で行い、act2のプレースに移動させる(ブロック897〜903)。この時、act2に移動させるクローンのマークド・インデックス595はact1に移動するクローンと異なり、「1」にしておく(ブロック899)。

【0142】act2に移動するクローンの作成及び移動の処理が終了すると、オリジン用モジュレーションモジュール605は、サクセッサ・リスト609を参照し、次のアクティビティが登録されているか否かを判断する(ブロック897)。この例の場合には、サクセッサ・リスト609には、もう登録されているアクティビティは存在しないので、オリジン用モジュレーション・モジュール605は、クローンの作成及び移動の処理が終了したことを認識する。

【0143】オリジン用モジュレーション・モジュール605はこれに応答して、モデレータ・エージェント参照モジュール599を介して、もとの場所(オリジンのプレース)に残っているモデレータ・エージェント351を消去するインストラクションをモデレータ・エージェント351に送出する(ブロック905)。

【0144】この一方、クローンのモデレータ・エージェントは、夫々のプレースにおいて、次のプランステップを実行する。このとき、カレント・ノードとしてANDSplitが再び取り出される。この次のカレントを決定するロジックのフローチャートを図31に示す。モジュレーション管理マネージャ603は、ステート管理部601を参照し、ステート情報を取得し、現在のステートはディスティネーションであることを認識し、ディスティネーション用モジュレーション607に処理の実行を依頼する。

【0145】モジュレーション管理マネージャ603はステートの情報をオリジンの状態に戻す。プランはANDSplitのディスティネーションの処理の終了を確認す

ると、マークド・インデックスに対応したプランノードを取り出し(act1またはact2)、アクティビティの実行を依頼する。

【0146】プラン・オブジェクト353はプラン・イタレータ352を参照して、次のカレントのノードを取り出す。このとき、プラン・イタレータは、サクセッサ・リスト609及びマークド・インデックス595を参照し、登録されている値を検査する(図31)。

【0147】act1、act2は、先に実行されたアクティビティが保持するResultを取得する。この例では、ANDSplitテンプレートの前にはアクティビティが実行されていないため、act1、act2は、先に実行されたアクティビティが保持するResultを、結果的に取得しないことになる。

【0148】まず、act1に対応するクローンのエージェントにおいて、次のカレント・ノードとしてANDJoinが選ばれる。プラン・オブジェクト353は、ANDJoinテンプレートのモデレータ管理マネージャ603に次のステップの実行を依頼する。モデレータ管理マネージャ603は、ステート管理部601の情報から、現在オリジンの処理を行うべきことを認識し、オリジン用モジュレーション・モジュール605に実行要求を出す。また、ステート管理部601にステートの変更を依頼する。

【0149】オリジン用モジュレーション・モジュール605はサクセッサ・リスト609を参照し、サクセッサ・リスト609に登録されたアクティビティ(act3)のアドレス情報を取得する。そして、モデレータ・エージェント参照モジュール599を介してモデレータ・エージェント351に入手したアドレス情報に対応したプレースへの移動を指示する。

【0150】act2に対応するクローンのエージェントもact1の場合と同様に処理を行い、act3のプレースに移動する。2つのモデレータ・エージェントは新たなプレースに到着したので、それぞれ次のステップを実行する。プラン・オブジェクト353は、カレント・ノードとしてANDJoinテンプレートを取り出し、次のステップを実行する。

【0151】図32、33は、ANDJoinテンプレートのディスティネーションにおける処理手順を示すフローチャートである。ANDJoinのモジュレーション管理マネージャ603は、ステート管理部601を参照して、現在のステートがディスティネーションであることを判断し、ディスティネーション用モジュレーション・モジュール607に実行を依頼する。

【0152】各モジュレーション管理マネージャ603は、ステートをオリジンの状態に戻す。act1に対応するクローンのディスティネーション用モジュレーション・モジュール607は、プレデセッサ・リスト591を参照し、終了のマークのついたアクティビティ(この場合a

act1)を取り出す(ブロック913)。この取り出されたアクティビティがプレデセッサ・インデックス・リストの中で一番小さなインデックスを有しているか否か検査し(ブロック915)、一番小さなインデックスを有していると判断された場合には、他のクローンエージェントのResultを受け取る処理を行う。

【0153】具体的には、ディスティネーション用モデレーション・モジュール607は、同一プレース上エージェント参照モジュール597に同一プレースに存在するエージェントリストをプレースから取得することを依頼する。同一プレース上エージェント参照モジュール597はこれに応答して、プレース360に対し"getAgentList"のインストラクションを出し、問い合わせを行う。プレースはAgentListを返すことによって、同一プレースに存在するエージェントのリストを入手する。

【0154】同一プレース上エージェント参照モジュール597はこのリストをディスティネーション用モデレーション・モジュール607に渡す。ディスティネーション用モデレーション・モジュール607は、このリストから同一のプランIDを有するエージェントを探す(ブロック921)。

【0155】同一のプランIDを有するエージェントが発見された場合、そのエージェントのプランを取得する。そして、取得したプランのカレント・ノードを調べ(ブロック929)、同じANDJoinであれば、そのエージェントのResultを受け取る処理を行う。本発明の好適な実施例においては、さらに、カレント・ノード・インデックスが同じか否かを判断し(ブロック931、933)、同じ場合のみ融合の処理に入る。これは、ANDJoinの処理対象を誤認することを防止するためである。このとき、ディスティネーション用モデレーション・モジュール607は、プレデセッサ・リストを参照し、プレデセッサが幾つあるかを把握し、その数に対応してResultを受け取る処理を行う。

【0156】具体的には、ディスティネーション用モデレーション・モジュール607は、受領したプランのプレデセッサ・リスト591を参照し、終了のマークのついたアクティビティを取り出し、そのアクティビティの保有しているResultを受け取る。そして、このResultを対応するアクティビティのResultにセットする。プランノード自体を置き換える処理を行ってもよい(ブロック935)。図34は、このResultを受け取る処理の概要を示す図である。

【0157】ディスティネーション用モデレーションエージェント607は、Resultのセットが終了したことを検出した後、モデレータ・エージェント参照モジュールを介して、相手のモデレータ・エージェント351に対して消去を指示するインストラクションを送出する(ブロック937)。相手のモデレータ・エージェントは、このインストラクションを受け、モデレータ・エ

ージェント及び、自分の管理するオブジェクト(プラン、テンプレート、アクティビティ等)を消去する。

【0158】この一方、act2に対応するエージェントは、新たなプレースに到着したことに伴い、次のステップを実行する。このエージェントのディスティネーション用モデレーション・モジュール607も、プレデセッサ・リスト591を参照し、終了のマークのついたアクティビティ(この場合act2)を取り出す(ブロック913)。この取り出されたアクティビティがプレデセッサ・インデックス・リストの中で一番小さなインデックスを有しているか否かを検査し(ブロック915)、一番小さなインデックスでないと判断される。この場合には、他のクローンエージェントのResultを渡すのを待ためSleep状態に入る(ブロック925)。

【0159】再び、話をact1に対応するクローンに戻すと、ディスティネーション用モデレーション・モジュール607は、プレデセッサ・リストの数に対応して(プレデセッサ・リストの要素数-1回)Resultを受け取り、クローンの消去を行う処理を完了した後、プランは、次のカレントのプランノードを取り出し(act3)、アクティビティの実行を依頼する。

【0160】act3は、act1、act2が内部に保持するresultを取得する。本発明の好適な実施例においては、ディスティネーションモジュール607がプレデセッサ・リスト591を参照し、(act1に対応するクローンの)act1、act2に対し、処理結果の送出を依頼する"getResult"のインストラクションを送出する。act1、act2は、これに応答し、自己の保有するResultを送出する。

【0161】act3は、内部に保持しているエージェントのクラス名をキーに同一プレースに存在するエージェント(この例ではレポートエージェント)の参照を得る。act3はレポートエージェントに対し内部に保持しているメッセージをact1、act2から入手したresultを引き数にして送る。

【0162】プランは次のカレント・ノードを取り出す。この場合、カレント・ノードは存在しないので、プランはこれを検出して処理を終了する。そして、モデレータ・エージェントは、処理の終了を検出しモデレータ・エージェン及び自己が管理するオブジェクトの消去をして処理を終了する。

【0163】H-4. ORSplitテンプレートを含むトポロジーの場合

図34は、ORSplitテンプレートによって制御される移動オブジェクト群の動作の概要を示す図である。ORSplitの場合、ChainやANDSplitの場合と異なりオリジンの処理において図35に示す処理を行う。ORSplitテンプレートは一つのアプレデセッサのアクティビティと複数のサクセッサのアクティ

ビティを有している。オリジナルのプレースにおいては、“chooseActivity”というメソッドが呼び出されることによってサクセッサのリストから一つのアクティビティが選択され、その選択されたアクティビティのみが実行される。

【0164】ORSplitのオリジンの処理においては、まず、ステートがディスティネーションに変更される（ブロック982）。そして、“chooseActivity”関数をコールし、所定の条件に合致する実行すべきアクティビティを選択する（ブロック983）。そして、そのアクティビティのインデックスをマークド・インデックスにする（ブロック984）。そして、選択されたアクティビティのアドレスを入手し、モデレータ・エージェント351に該アドレスへの移動を要求する（ブロック985）。

【0165】H-5. ORJoinテンプレートを含むトポロジーの場合

図36は、ORJoinテンプレートによって制御される移動オブジェクト群の動作の概要を示す図である。ORJoinの場合、ChainやANDJoinの場合と異なりディスティネーションの処理において図37に示すような処理を行う。ORJoinテンプレートは、複数のプレデセッサのアクティビティと一つのサクセッサのアクティビティを有している。ORJoinは、ディスティネーションにおいて、最も早く到着したものがDisposeエージェントを生成し、他の移動エージェントを消去する処理を行う。

【0166】まず、最も早くディスティネーション690に到着した移動オブジェクト群のORJoinテンプレート694は、“findDisposeAgent”のメソッドを送出して、同一プレース上にDisposeエージェントが存在しているか否かを判断する（ブロック943）。存在しない場合は、他のエージェントを消去するDisposeエージェントを生成する（ブロック945）。このとき引き数としてプラン及びプレデセッサ・リストのアクティビティの数-1を渡す。Disposeエージェントは、プレデセッサ・リストのアクティビティの数-1の移動オブジェクト群を待ち受け消去し、役目が終了すると自分自身を消去する。本発明の好適な実施例においては、最も早くディスティネーション690に到着したか否かを生き残りの条件としているが、Resultの内容を条件として、継続処理するものを決定してもよい。

【0167】この一方、他の移動オブジェクト群のORJoinテンプレート696は、同様に“findDisposeAgent”のメソッドを送出して、同一プレース上にDisposeエージェントが存在しているか否かを判断する（ブロック943）。この場合、既にDisposeエージェントが存在しているので、Disposeエージェントに消去される処理に入る。具体的には、Dispos

eエージェントが同じプランを保持するか（ブロック947）、また、同じカレント・ノード・インデックスか（ブロック949）判断し、自分が消去されるべき、Disposeエージェントであると確認する。

【0168】自分が消去されるべき、Disposeエージェントであると確認できた場合には、Disposeエージェントに“countdown”のメッセージを送り、Disposeエージェントの処理数を減らす（ブロック951）。そして、ORJoinテンプレート696は、モデレータ・エージェント692にdispose要求を送出し、モデレータ・エージェント692を消滅させる。

【0169】その他、本発明の好適な実施例においては、複数のプレデセッサのアクティビティと、プレデセッサのアクティビティよりも少ない数の複数のサクセッサのアクティビティを有する一部Joinテンプレート等も提供されているが、これらのテンプレートは上述のテンプレートの組合せでも実施可能であるため説明を省略する。例えば、一部Joinテンプレートは複数のプレデセッサのアクティビティと、1つのサクセッサのアクティビティとをリンクするORJoinテンプレートと、所望の数のORJoinテンプレートが実行されたことを判断するアクティビティと、このアクティビティが所望の条件をクリアしたか否かの判断によってORJoinテンプレートに戻るIterationテンプレートと、ダミーのアクティビティと、所望の数に分裂させるANDSplitの組合せによって実施可能である。

【0170】

【発明の効果】以上説明したように、本発明によれば、高度なプログラミングの知識を必要とせず、プログラムの開発労力及び開発時間の負担が少ない移動エージェントの開発が可能となる。

【0171】また、本発明によれば、移動エージェントの開発労力及び開発時間を可能な限り軽減することができる。

【0172】そして、本発明によれば、開発者が直感的にその機能を把握でき、親しみやすい移動エージェント開発環境を提供することができる。

【0173】

【図面の簡単な説明】

【図1】 本発明によって生成される移動エージェントが動作する分散ネットワーク環境を示す図である。

【図2】 本発明によって生成される移動エージェントが分散ネットワーク上で移動する態様の一例を示す図である。

【図3】 本発明の好適な実施例における移動エージェント生成システムのハードウェア構成の概観図である。

【図4】 本発明の移動エージェント生成システムにおける処理要素の一実施例を示す機能ブロック図である。

【図5】 本発明の移動エージェント生成システムにおける処理要素の一実施例をGUIプラン定義データ生成部207を中心に見た機能ブロック図である。

【図6】 本発明の移動エージェント生成システムによって生成された移動エージェント群の実行時における処理要素の一実施例を示す機能ブロック図である。移動オブジェ

【図7】 本発明の好適な実施例において生成される移動エージェント群のオブジェクト図である。

【図8】 本発明の好適な実施例において生成される移動エージェント群のオブジェクト図である。

【図9】 本発明の好適な実施例において生成されるプラン定義データのオブジェクト図である。

【図10】 本発明の好適な実施例におけるオブジェクト間のメッセージの流れを示す図である。

【図11】 本発明の好適な実施例における移動エージェント生成システムのユーザインターフェースを示す図である。

【図12】 本発明の好適な実施例における移動エージェント生成システムのユーザインターフェースを示す図である。

【図13】 本発明の好適な実施例における移動エージェント生成システムのユーザインターフェースを示す図である。

【図14】 本発明の好適な実施例における移動エージェント生成システムのユーザインターフェースを示す図である。

【図15】 本発明の好適な実施例における移動エージェント生成システムのユーザインターフェースを示す図である。

【図16】 本発明の好適な実施例における移動エージェント生成システムのユーザインターフェースを示す図である。

【図17】 本発明の好適な実施例におけるプラン定義データを基に実行コード（プラン）を生成する手順を示すフローチャートである。

【図18】 本発明の好適な実施例におけるプラン定義データを基に実行コード（プラン）を生成する手順を示すフローチャートである。

【図19】 本発明の好適な実施例におけるキャリア・エージェントの生成及び各プレースへの送信手順を示すフローチャートである。

【図20】 本発明の好適な実施例におけるキャリア・エージェントの生成及び各プレースへの送信手順を示すフローチャートである。

【図21】 本発明の好適な実施例におけるプラン・オブジェクトの実行手順を示すフローチャートである。

【図22】 本発明の好適な実施例におけるモデレータ・テンプレートの機能ブロック図である。

【図23】 本発明の好適な実施例におけるプレースの

概念図である。

【図24】 本発明の好適な実施例におけるモデレータ・テンプレートの処理概要を示すフローチャートである。

【図25】 本発明の好適な実施例における移動オブジェクト群がオリジンのプレースからディスティネーションのプレースに移動する状態を示す概念図である。

【図26】 本発明の好適な実施例における移動オブジェクト群がオリジンのプレースからディスティネーションのプレースに移動する状態を示す概念図である。

【図27】 本発明の好適な実施例におけるIterationテンプレートのオリジンにおける処理手順を示すフローチャートである。

【図28】 本発明の好適な実施例における移動オブジェクト群がオリジンのプレースからディスティネーションのプレースに移動する状態を示す概念図である。

【図29】 本発明の好適な実施例における移動オブジェクト群がオリジンのプレースからディスティネーションのプレースに移動する状態を示す概念図である。

【図30】 本発明の好適な実施例におけるAND Splitテンプレートのオリジンにおける処理手順を示すフローチャートである。

【図31】 本発明の好適な実施例におけるカレント・ノードを決定するロジックを示すフローチャートである。

【図32】 本発明の好適な実施例におけるAND Joinテンプレートのディスティネーションにおける処理手順を示すフローチャートである。

【図33】 本発明の好適な実施例におけるAND Joinテンプレートのディスティネーションにおける処理手順を示すフローチャートである。

【図34】 本発明の好適な実施例における移動オブジェクト群がオリジンのプレースからディスティネーションのプレースに移動する状態を示す概念図である。

【図35】 本発明の好適な実施例におけるOR Splitテンプレートのオリジンにおける処理手順を示すフローチャートである。

【図36】 本発明の好適な実施例における移動オブジェクト群がオリジンのプレースからディスティネーションのプレースに移動する状態を示す概念図である。

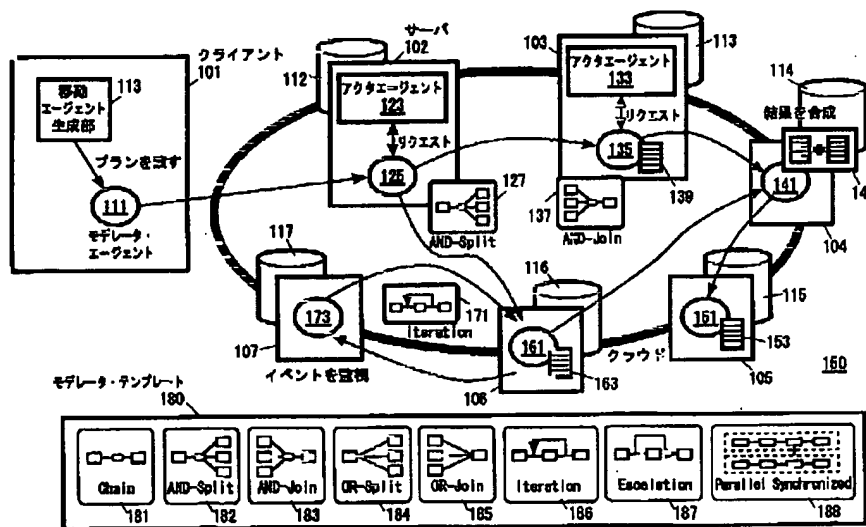
【図37】 本発明の好適な実施例におけるOR Joinテンプレートのディスティネーションにおける処理手順を示すフローチャートである。

【符号の説明】

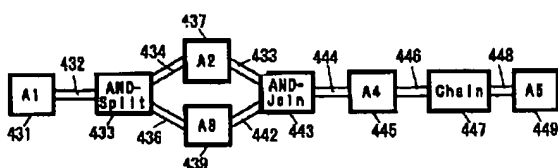
- 100 ノード・システム
- 201 GUIプランノード・ライブラリ
- 203 GUIモデレータ・テンプレート
- 205 GUIアクティビティ
- 207 GUIプラン定義データ
- 209 GUIプラン定義データ生成部

- | | |
|-------------------------|-------------------------|
| 211 実行コード生成部 | 257 キャリア・エージェント |
| 213 プラン生成部 | 261 プラン・オブジェクト |
| 215 プランID生成部 | 263 プラン実行部 |
| 221 プラン・オブジェクト | 265 順序リスト |
| 222 プロキシ・アクティビティ | 267 カレント・ノード |
| 223 プラン構成部 | 269 コンクリート・アクティビティ・テーブル |
| 224 プラン分割部 | 270 移動オブジェクト群本体 |
| 225 順序リスト | 271 モデレータ・テンプレート |
| 227 プランID | 273 モデレータ・エージェント制御部 |
| 229 ネットワーク・トポロジー | 275 プラン・ステップ実行部 |
| 230 キャリア・エージェント | 277 アドレス参照部 |
| 231 入力イベント取得部 | 280 キャリア・エージェント |
| 232 移動先プレース・アドレス | 281 プロキシ・アクティビティ |
| 233 プラン・オブジェクト操作部 | 282 コンクリート・アクティビティ |
| 234 キャリア・エージェントID | 283 プロキシ・アクティビティ実行部 |
| 236 コンクリート・アクティビティ・テーブル | 284 コンクリート・アクティビティ実行部 |
| 237 プラン・オブジェクト属性変更部 | 285 メッセージ機能 |
| 243 プラン表示部 | 286 メッセージ機能 |
| 249 実行コード生成部 | 291 アクタ・エージェント |
| 251 モデレータ・エージェント | 293 メッセージ機能 |
| 253 移動エージェント・ライフサイクル制御部 | 295 メッセージ処理部 |
| 255 プラン実行部 | |

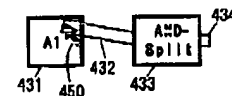
【図1】



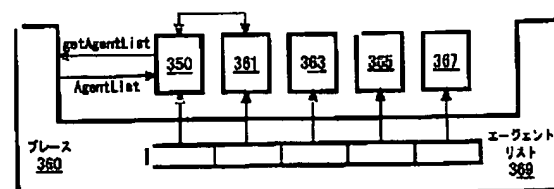
【図15】



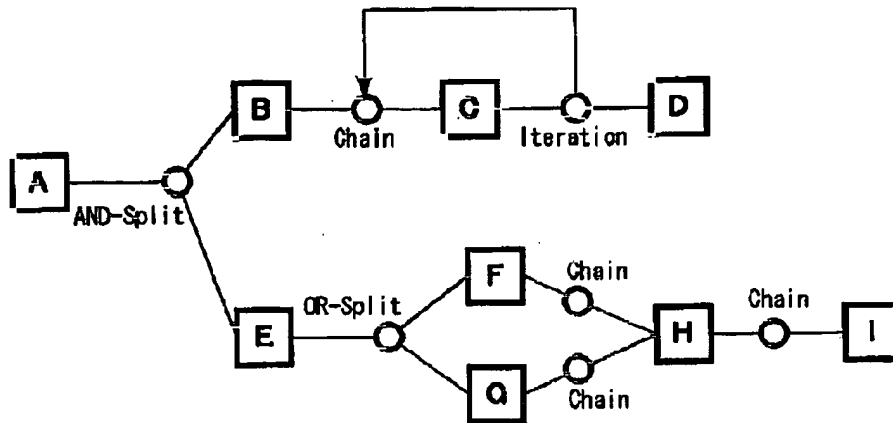
【図14】



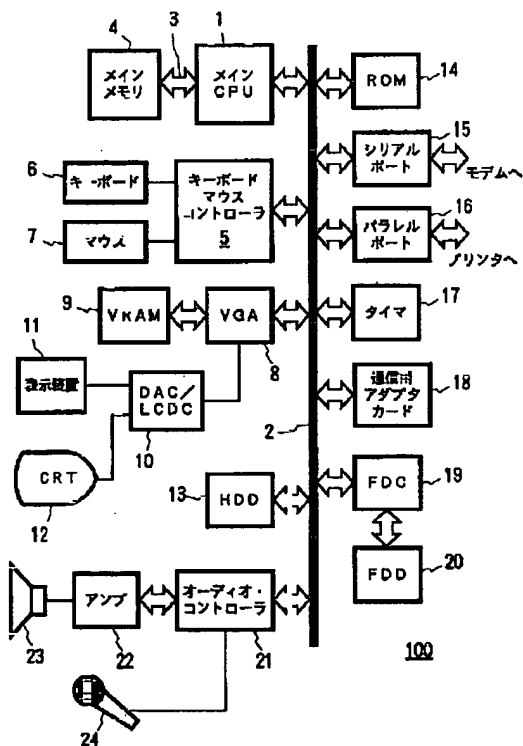
【図23】



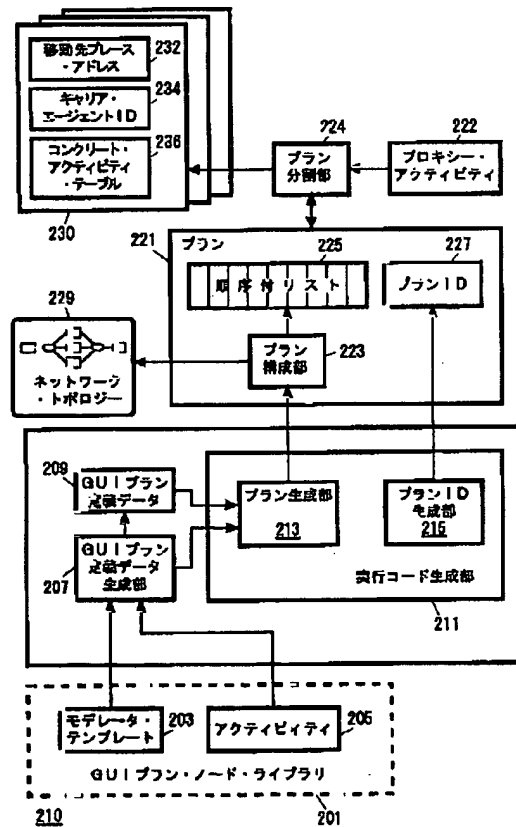
【図2】



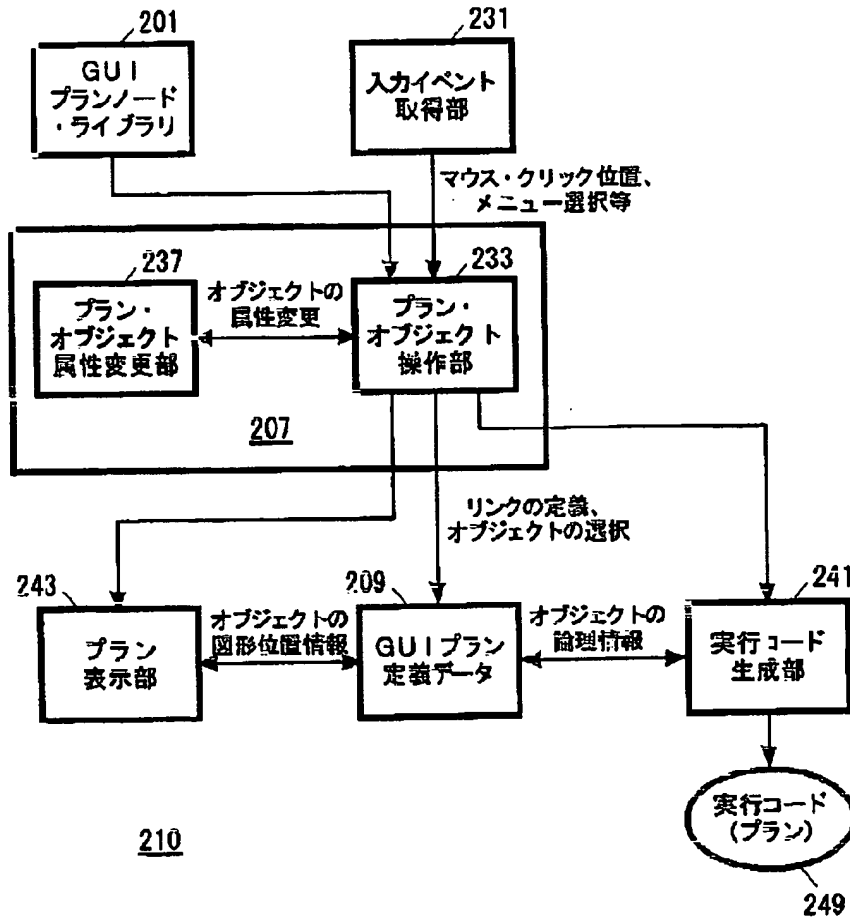
【図3】



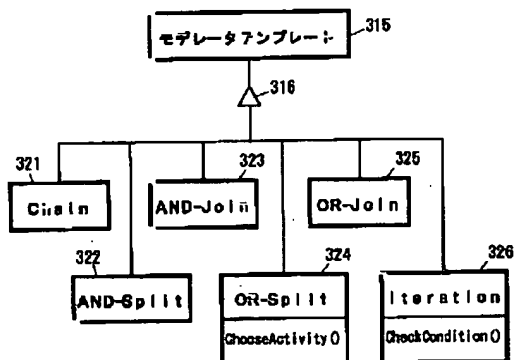
【図4】



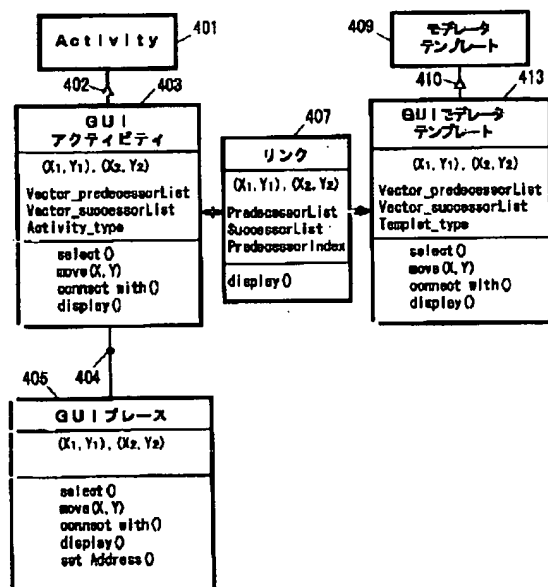
【図5】



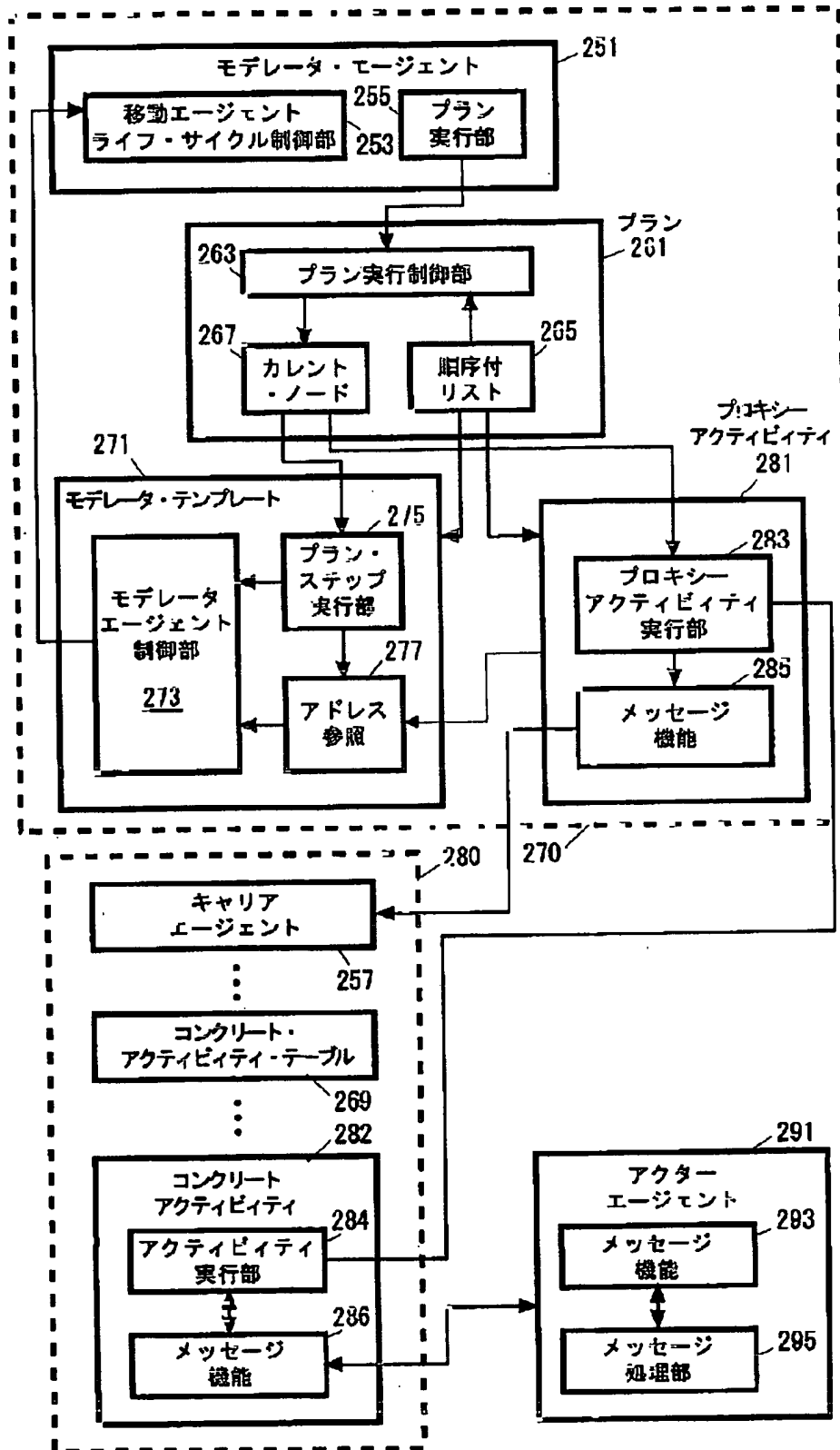
【図8】



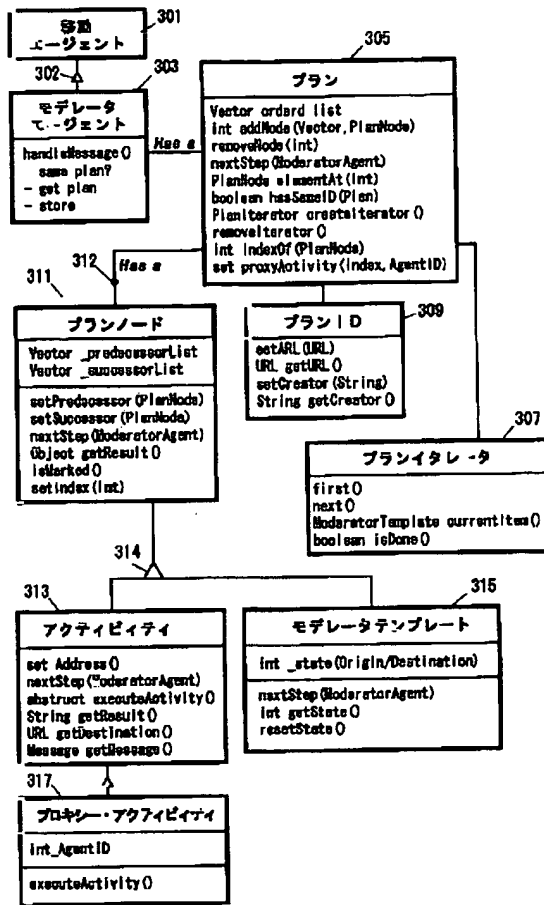
【図9】



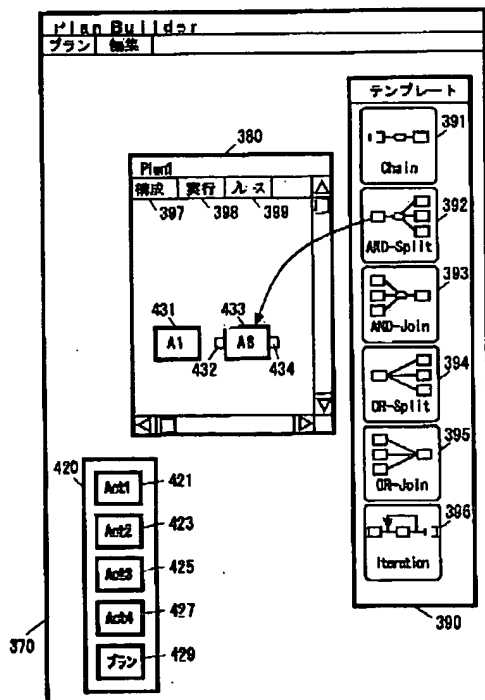
【図6】



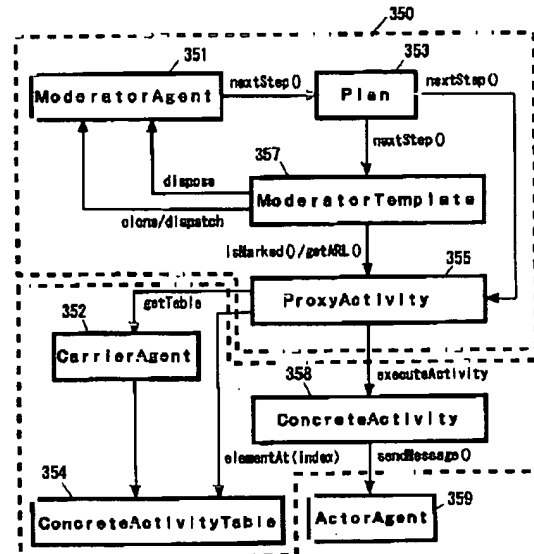
【図7】



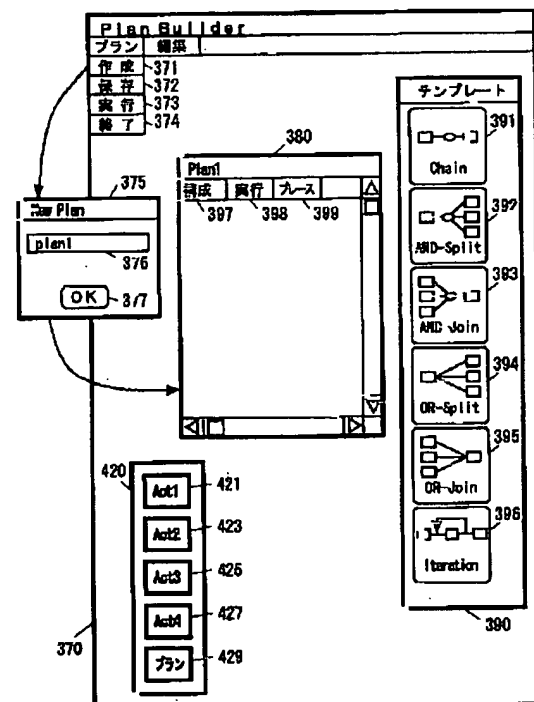
【図13】



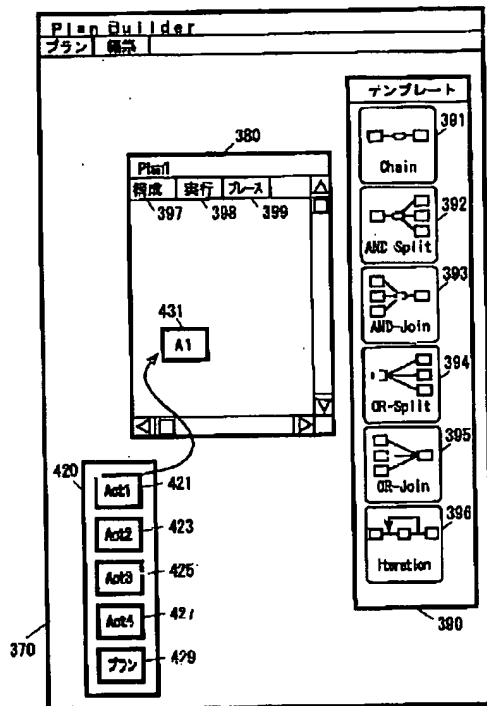
【図10】



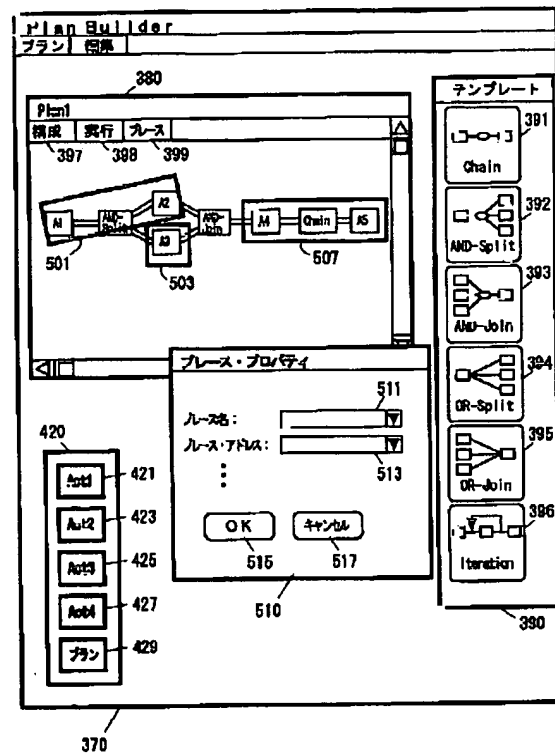
【図11】



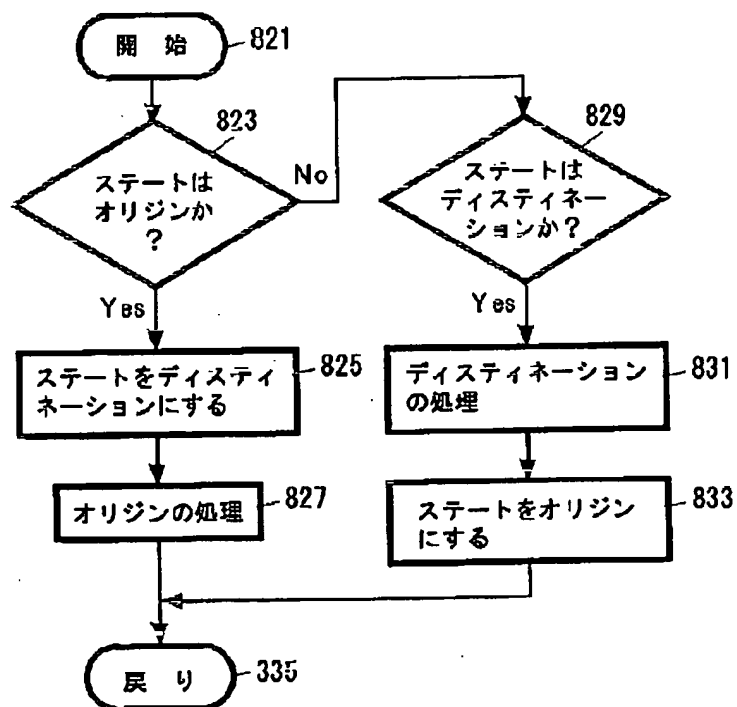
【図12】



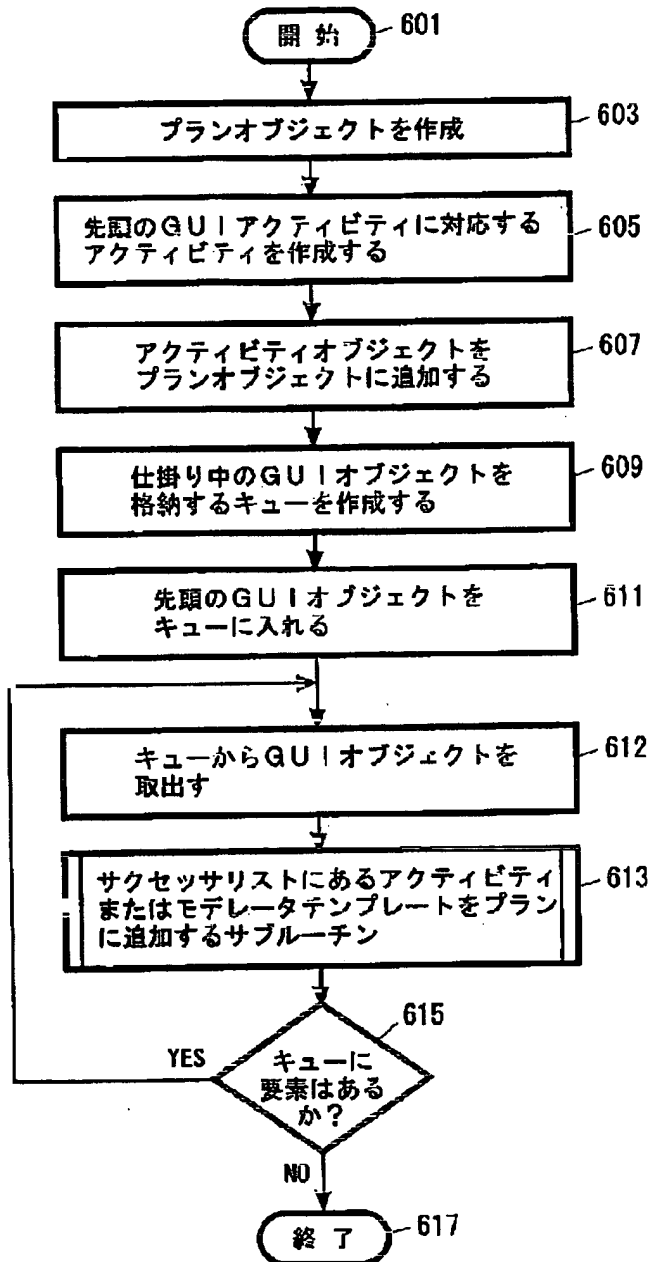
【図16】



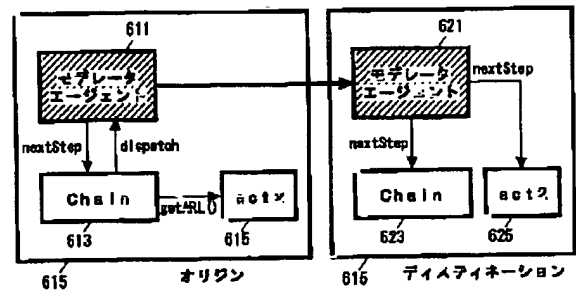
【図24】



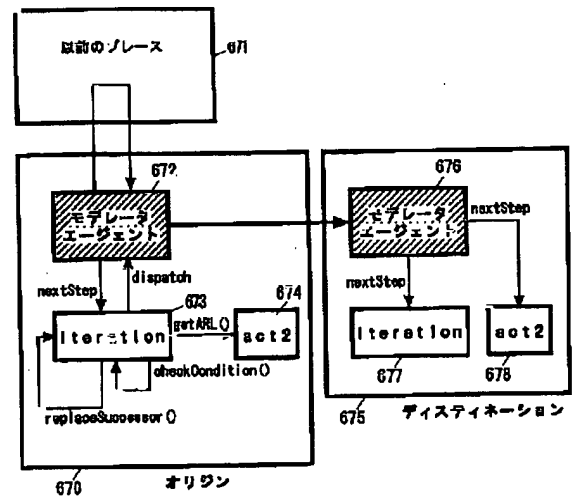
【図17】



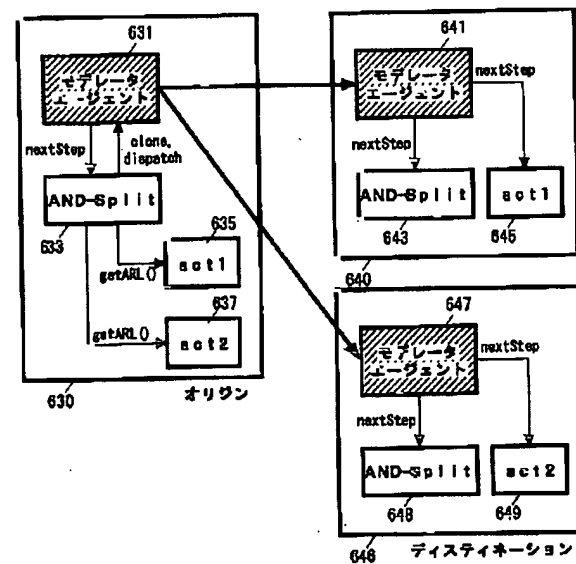
【図25】



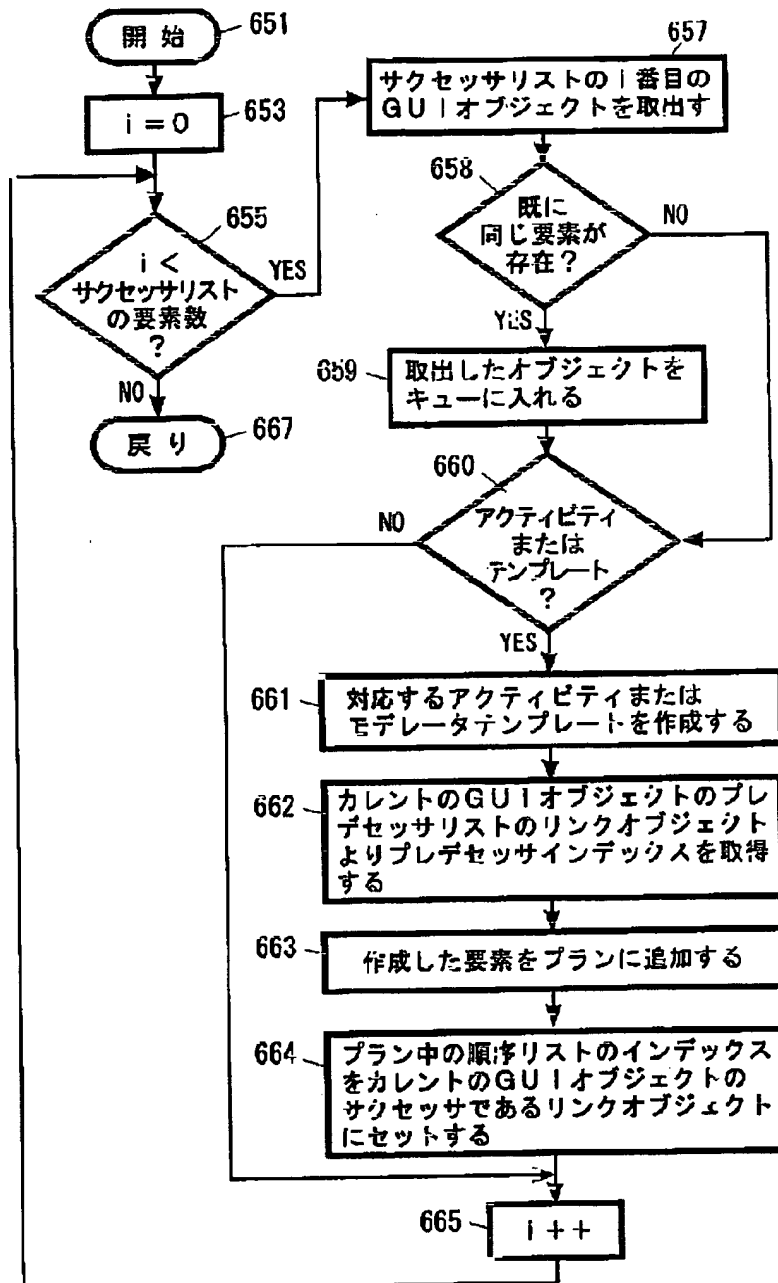
【図26】



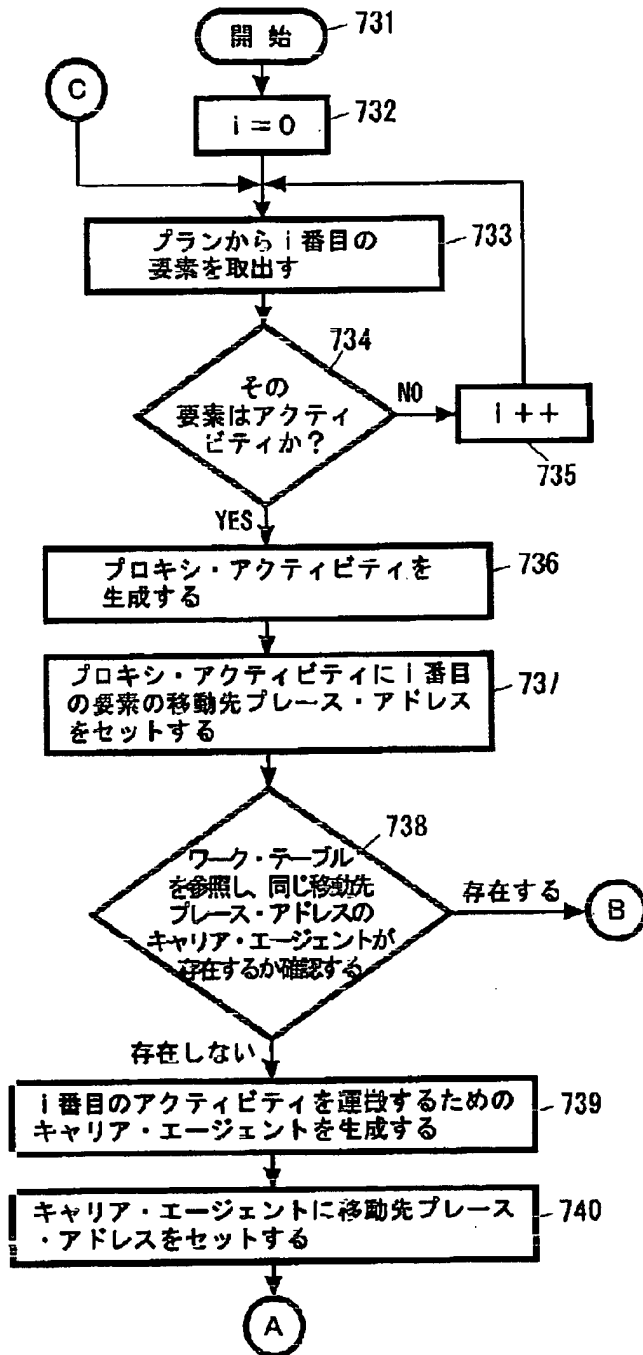
【図28】



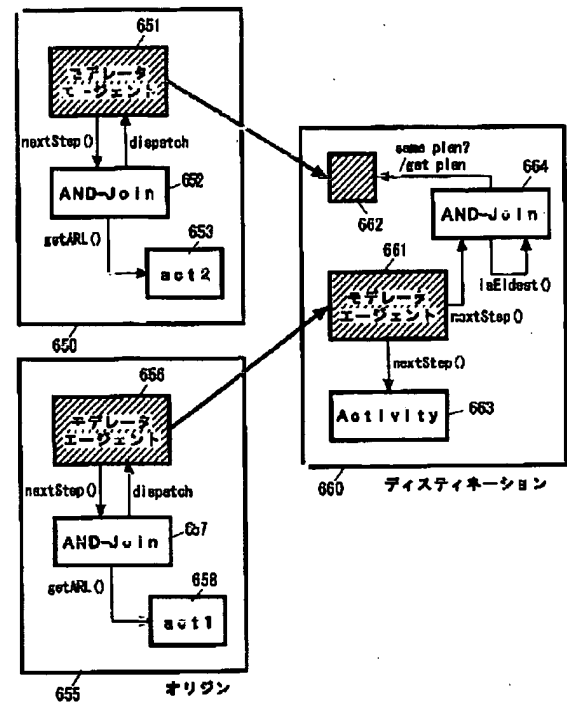
【図18】



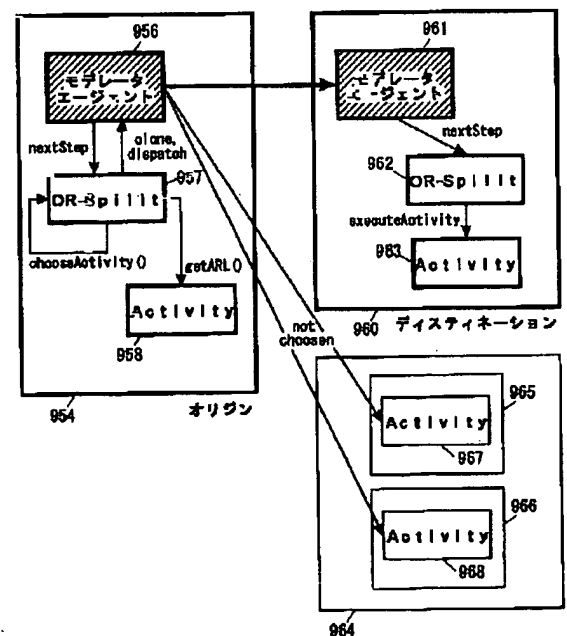
【図19】



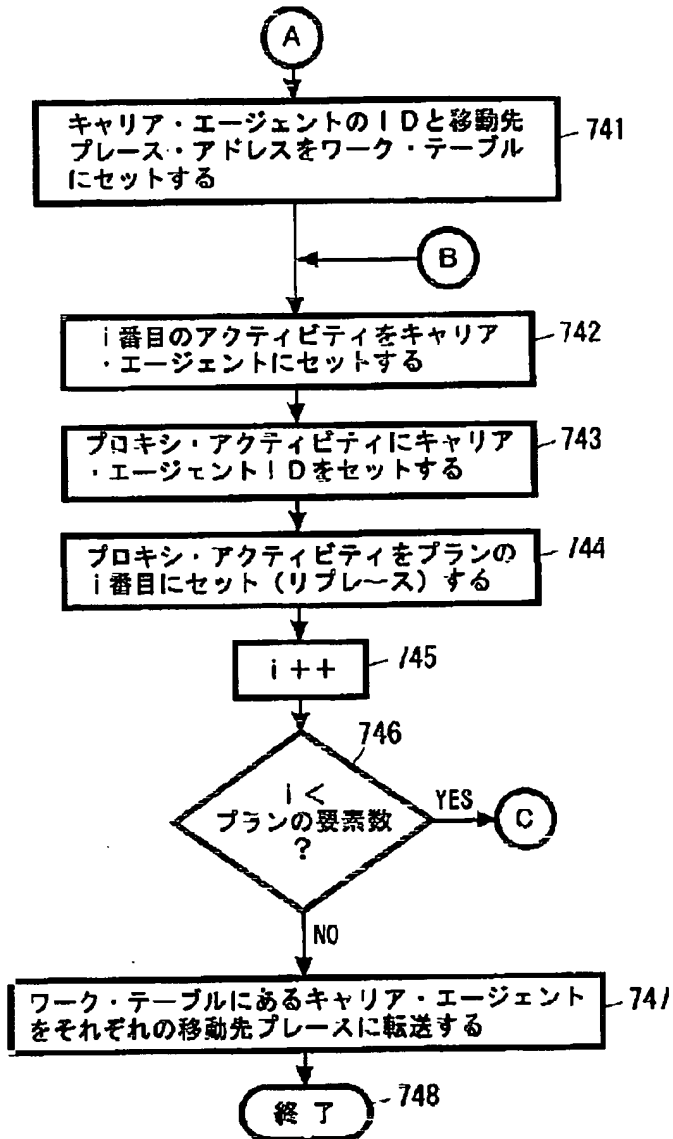
【図29】



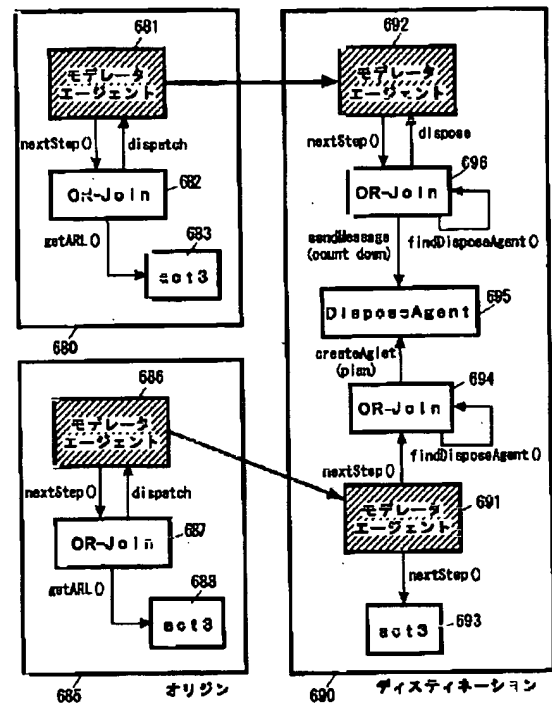
【図34】



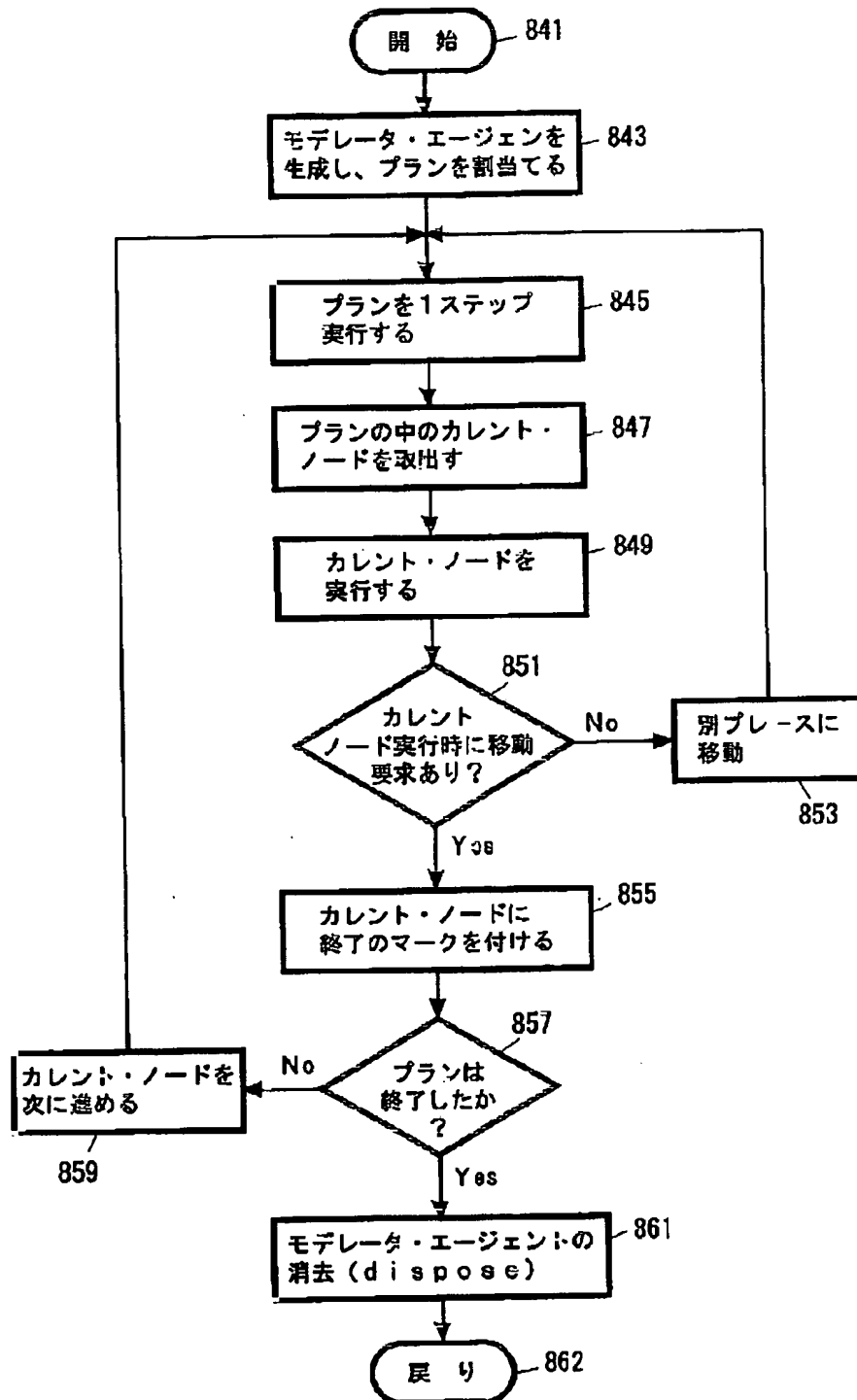
【図20】



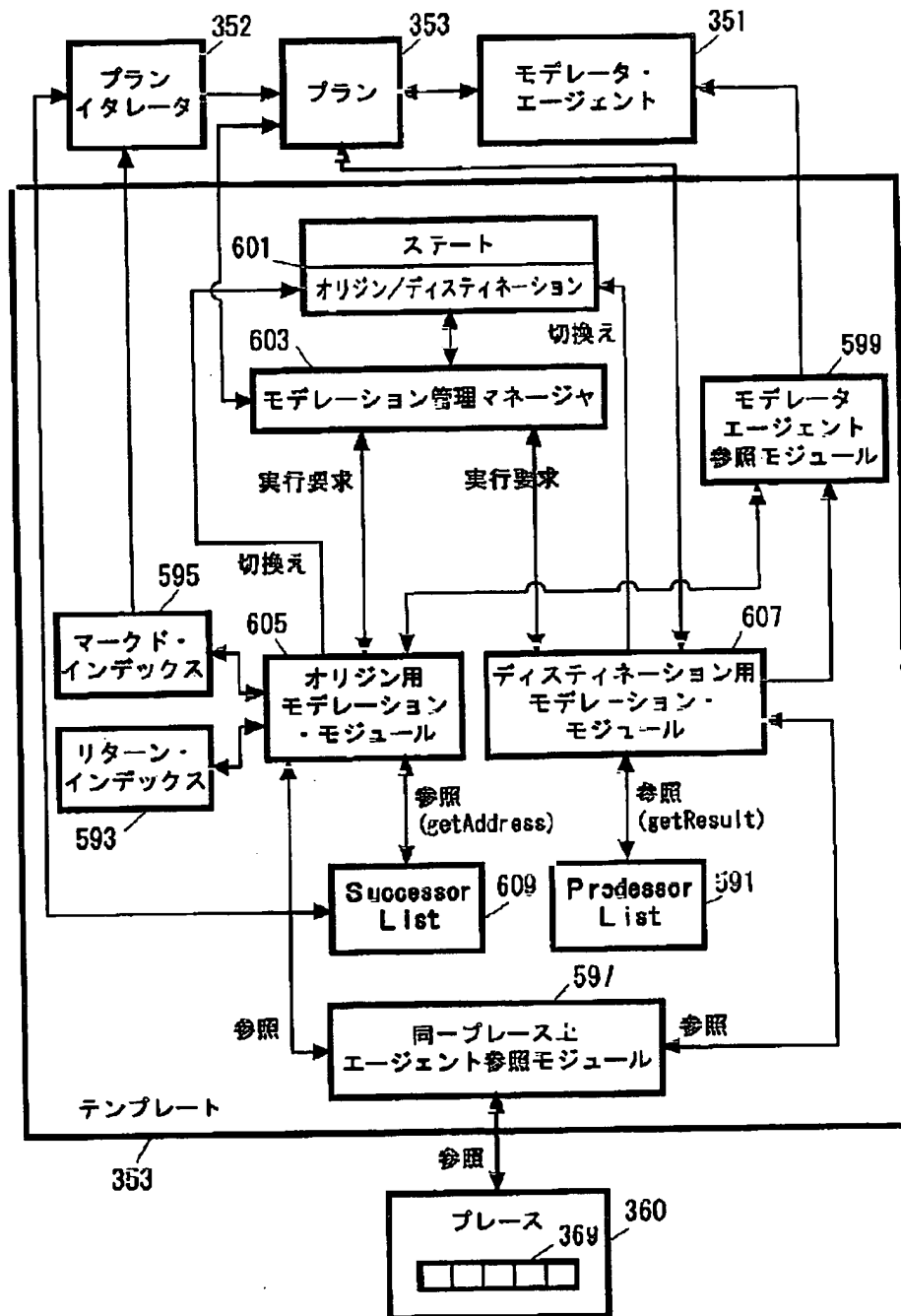
【図36】



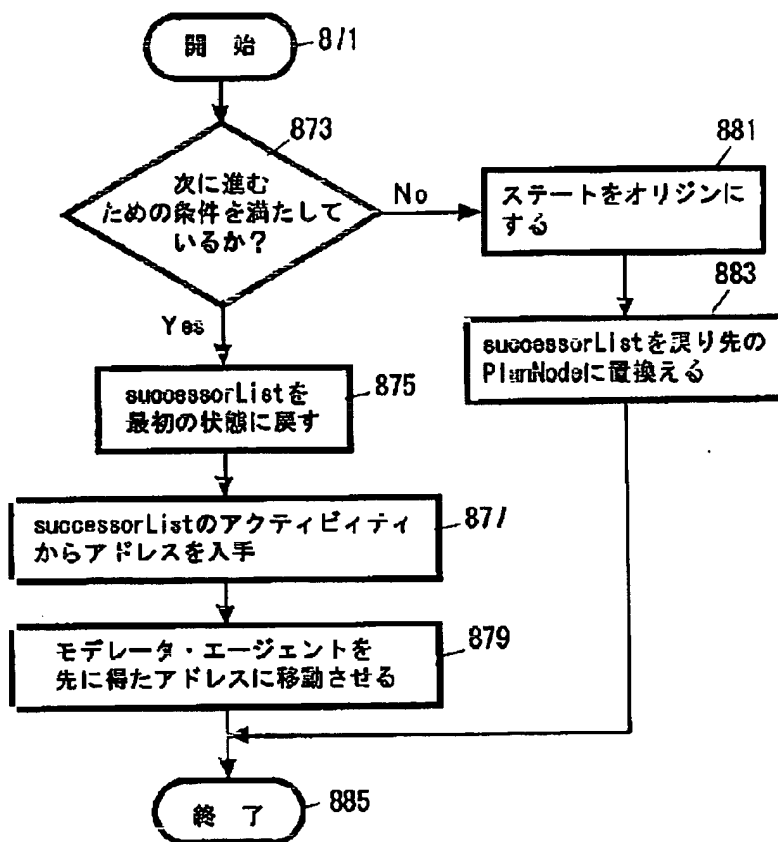
【図21】



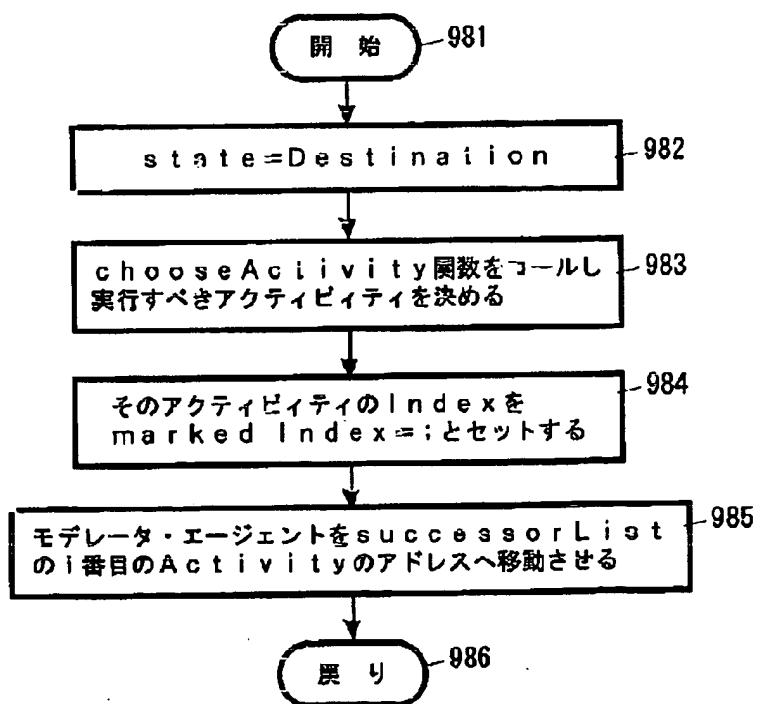
【図22】



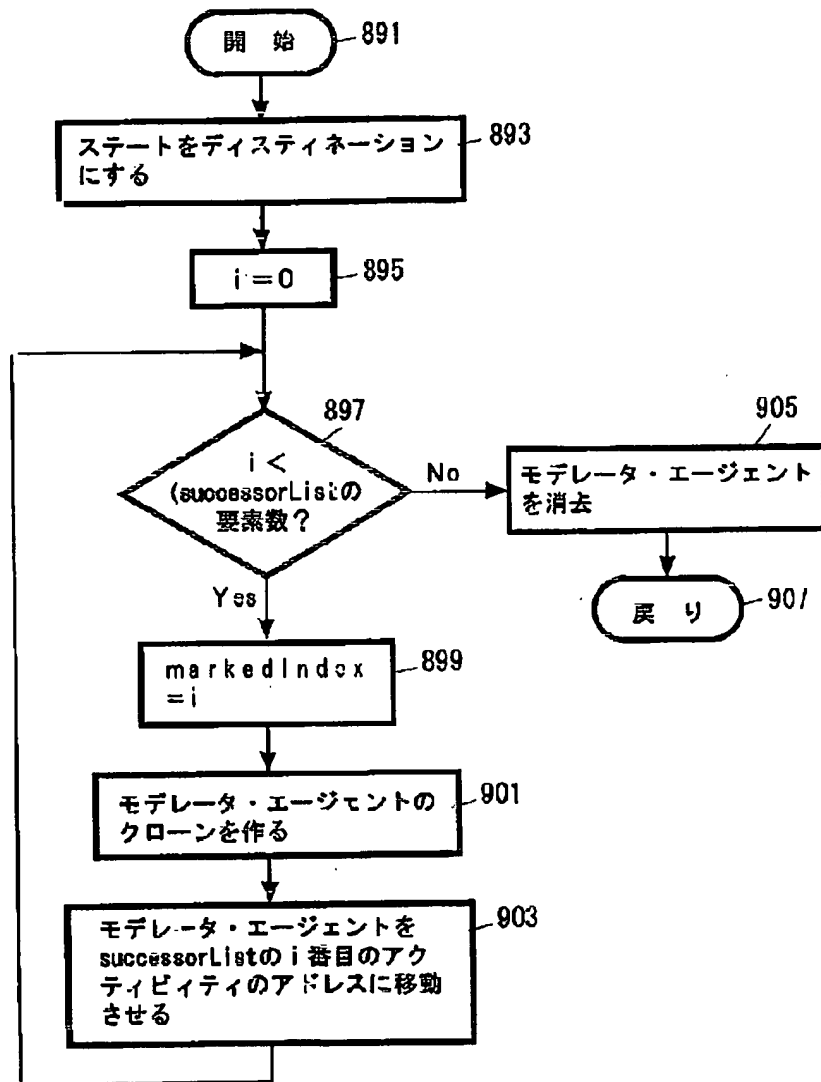
【図27】



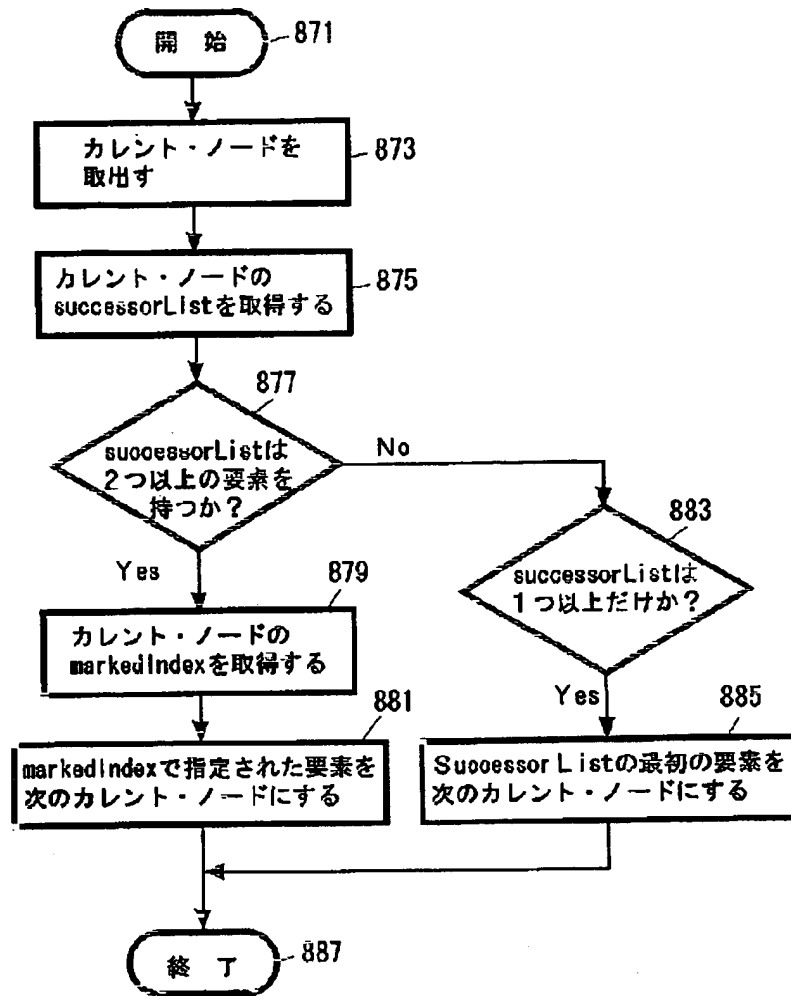
【図35】



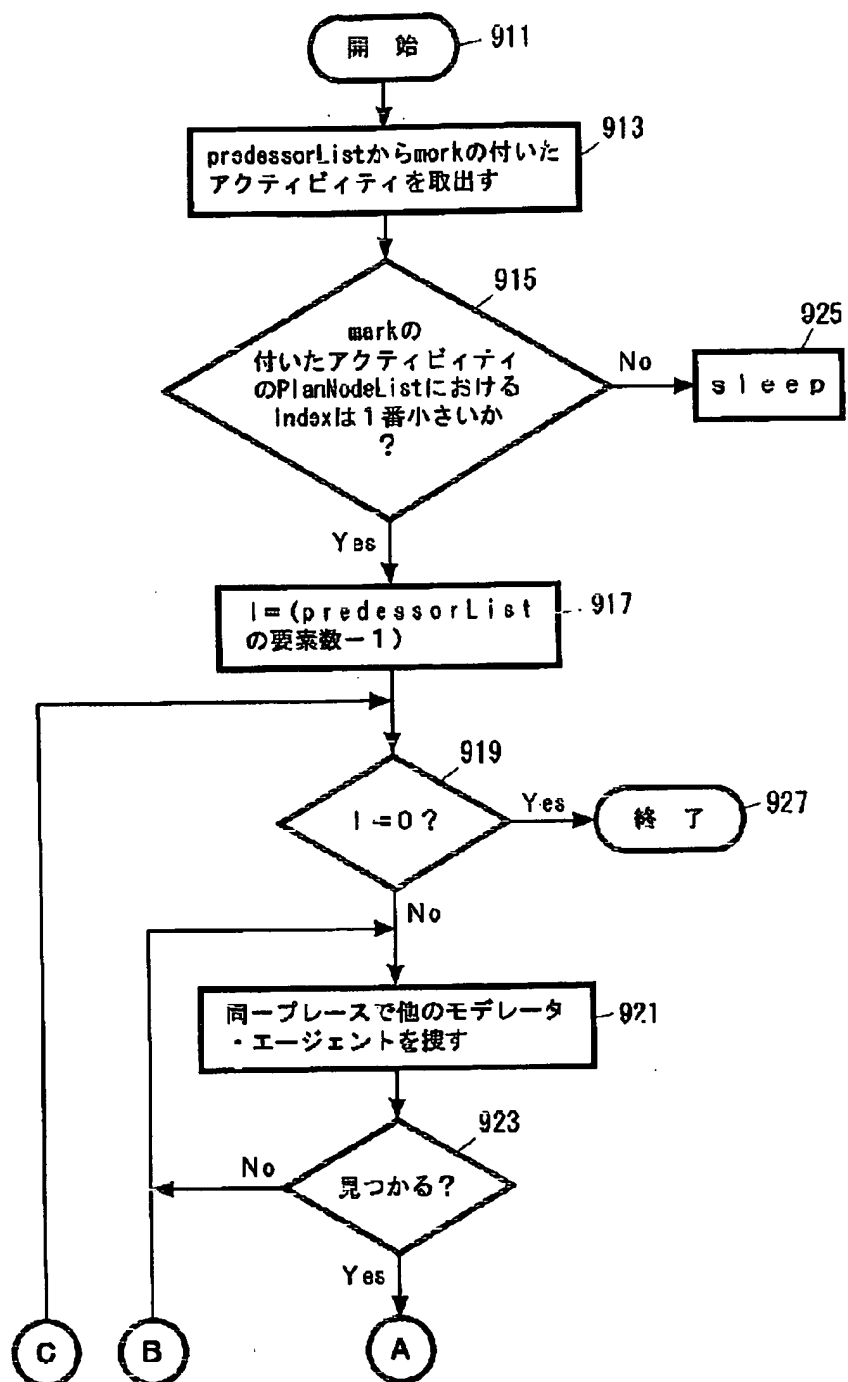
【図30】



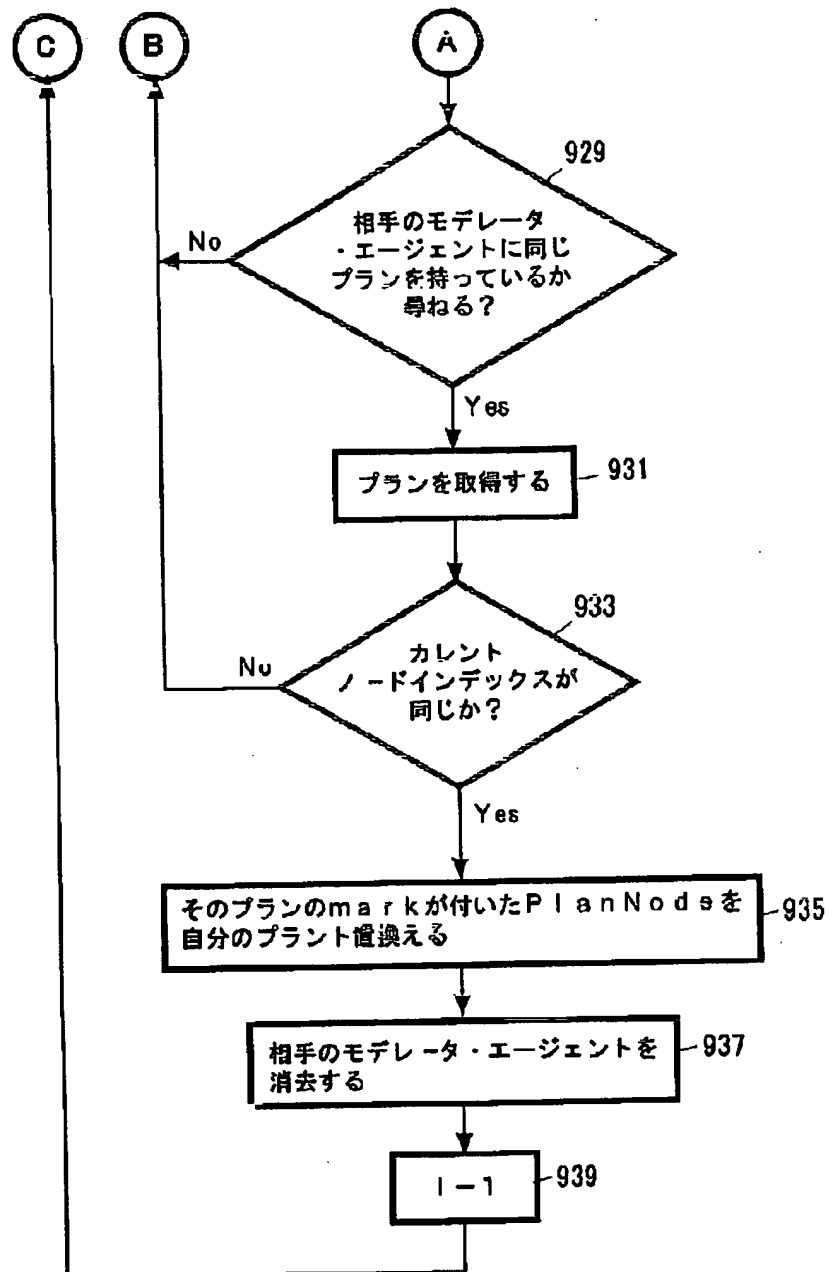
【図31】



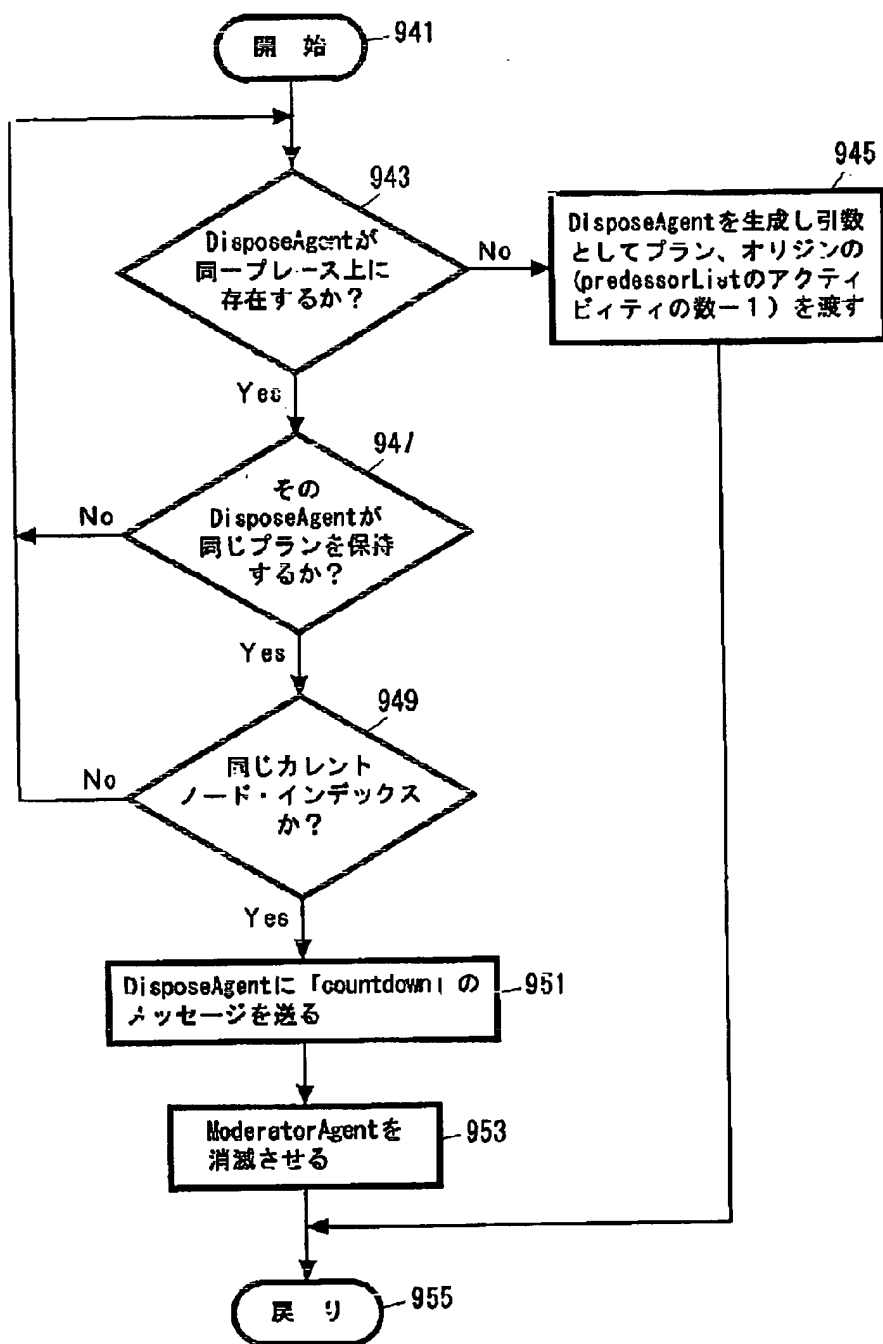
【図32】



【図33】



【図37】



フロントページの続き

(72)発明者 南 和 宏

神奈川県大和市下鶴間1623番地14 日本ア
イ・ビー・エム株式会社 大和事業所内